Low-dimensional Models for 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 often collect traffic information with high temporal and spatial resolutions. The volume of the collected data severely limits the scalability of online traffic operations. To overcome this issue, we propose a low-dimensional network representation where only a subset of road segments is explicitly monitored. Traffic information for the subset of roads is then used to estimate and predict conditions of the entire network. Numerical results show that such approach provides 10 times faster prediction at a loss of performance of 3% and 1% for 5 and 30 minutes prediction horizons, respectively.**

***Index Terms*—Low-dimensional models, traffic prediction.**

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. Recorded data is frequently used for historical analysis and traffic management operations such as network monitoring, transportation planning and congestion avoidance applications [1]. These applications heavily rely on fast and accurate assessment of current (estimation) and future (prediction) network states.

To model the road network, existing studies explicitly address each road segment in that network. For large traffic networks and online applications such an approach may not be feasible. To overcome this problem we focus on low-dimensional network models where only a subset of road segments needs to be explicitly monitored.

In this study, we use column-based (CX) matrix decomposition to express the original network in terms of a small subnetwork. We refer to the small subnetwork as the compressed state of the original network. We learn the relationship between compressed and original (uncompressed) network by analyzing the recorded data in offline manner. In this way, we can represent the traffic network as a product of two low-rank matrices: (i) the subnetwork data and (ii) the corresponding relationship matrix. We refer to this as CX compression scheme. The CX compression scheme is a stepping stone to compressed sensing and compressed prediction applications.

In the case of compressed sensing we aim to infer the present state of the entire network from the current traffic state of the subnetwork [2]. We use the training data (collected offline) to infer the relationship matrix. To assess the network state we multiply the data from the subnetwork, obtained from the testing set, with the relationship matrix, inferred from the training data set. Our underlying assumption is that traffic variables often vary smoothly across the traffic network [3].

In the matter of compressed prediction we apply the CX-based method to infer the future state of the network. First, we explicitly predict traffic state for the subnetwork using traditional prediction algorithms. Then we multiply the predicted data of the subnetwork with the relationship matrix, inferred from the training data set. Similarly to compressed sensing we rely on the observation that traffic conditions tend to follow distinct patterns and traffic parameters often vary smoothly [3], [4].

For our analysis, we consider the city-scale traffic network in Singapore, comprising $17,967$ road segments. The numerical results show that the proposed methods can infer the current and future states of the network, while substantially improving the processing speed of the underlying modeling algorithm. The reduction in computational time 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 review relevant literature. In Section III we introduce the column based (CX) matrix decomposition method. In Section IV we present three applications of the CX matrix decomposition methods in the realm of traffic modeling: compression, compressed sensing, and compressed prediction. In Section V we describe the traffic data set analyzed in this paper. In Section VI we provide and discuss results for our experiments. In Section VII we summarize our contributions and suggest topics for future research.

II. Related work

In this paper we propose a novel low-dimensional network model to improve the scalability of estimation and prediction operations in ITS. Low-dimensional representation of large traffic data sets is traditionally obtained by Principal Component Analysis (PCA) [3], [5], [6]. PCA provides an

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).

effective low-dimensional representation in terms of latent variables and corresponding basis vectors. However, these latent variables are hard to interpret. Moreover, even if we obtain the basis vectors from historical data, we still need to collect data from all sensors during online operation. Due to this reason, PCA is often used for offline operations such as compression and data preprocessing [6].

Simulation (model) and data driven approaches are traditionally used to perform traffic estimation and prediction [7], [8]. Simulation approaches can be used for traffic management operations at various levels of network granularity [7], [9], [10]. For large areas, macroscopic and mesoscopic simulation tools (e.g., DYNAMIT) have been adopted to build custom models, relying on historical speed-density link relationships for that specific network [10]. In recent years, large volumes of collected data have served for extensive model calibration of traffic dynamics. Extensive calibration enhances the credibility of built simulation models. However, such models are not generic and cannot be translated from one network to another in a straightforward manner. An alternative is to consider data driven methods. These methods offer greater flexibility due to their generic structure. Consequently, these methods are used to develop highly accurate traffic estimation and prediction models [8], [11], [12]. In all of these studies, data-driven techniques explicitly predict traffic variables at each link in the observed network. 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 explicitly monitored.

III. Column based (CX) decomposition

The column based (CX) method has recently found applications in many fields such as text processing, finance and biology [13]–[15]; it uses only a subset of the columns to reconstruct the entire data matrix. In our previous study, we applied the column and row (CUR) based method to impute a matrix of traffic data from a few columns (links) and few rows (time instances) of that matrix [16]. Since the CUR method occasionally requires traffic data for the entire network, it cannot be applied for compressed prediction. The CX-based method instead does not have this requirement. In the following, we will briefly review the CX-based method.

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 $\mathbf{X} \in \mathbb{R}^{c\times n}$ is a matrix that expresses every column of $\mathbf{A}$ in terms of the basis provided by the columns of $\mathbf{C}$ [17].

A. Column selection

In order to select the best subset of columns, for a given size c, one needs to test all possible combinations. However, the computational complexity of this brute-force approach is $O(n^c)$ [13]. Due to this complexity, testing all possible choices of c columns is typically not practical. To alleviate this problem, several randomized algorithms have been proposed [17], [18]. In our numerical experiments, the SVD sampling method yields the best reconstruction accuracy [16]. The SVD sampling algorithm assigns higher selection probability to the road segments with larger traffic speed variations [16]. This algorithm calculates the Euclidean norm of top k right singular vectors of matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ to assign a score E_{a_i} to each column [17]. This score (E_{a_i}) is then converted into a probability P_{a_i} and further used to sample the columns:

$$\mathrm{P}_{a_i} = \frac{1}{k} E_{a_i} = \frac{1}{k}\sum_{j=1}^{k} v_{ij}^2 \quad \forall\ i = 1,...n, \tag{1}$$

where v_{ij} is the i-th coordinate of j-th right singular vector.

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 $\mathbf{C}$ [17]. The matrix $\mathbf{X}$ can be regarded as an extrapolation matrix that maps the subnetwork associated with $\mathbf{C}$ to the entire network represented by $\mathbf{A}$. For given matrices $\mathbf{C}$ and $\mathbf{A}$, we compute the matrix $\mathbf{X}$ as $\mathbf{X} = \mathbf{C}^{+}\mathbf{A}$, where $\mathbf{C}^{+}$ is Moore-Penrose pseudo-inverse of matrix $\mathbf{C}$ [19].

IV. CX-based method for traffic applications

In this section, we discuss how CX based method can be used to perform compression, compressed sensing, and compressed prediction of traffic 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 $\mathbf{A}$ contains the traffic state 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 $\mathbf{A}$ contains the observed condition at location s_j during the entire recording period. Hence, we can write traffic data matrix as $\mathbf{A} = [\mathbf{a}_1 \ldots \mathbf{a}_n]$. 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

Suppose that the matrix $\mathbf{C}_h$ contains the observed traffic states of the 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 $\hat{\mathbf{A}}_h = \mathbf{C}_h\mathbf{X}_h$, where the matrix $\mathbf{X}_h$ contains the relationships between the traffic condition 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:

$$CR_h = \frac{mn}{mc + cn}. \tag{2}$$

Column based (CX) compression scheme leads to simple network representation. Although such compression scheme does not outperform PCA, still it could be useful for online traffic monitoring operations [16]. In the following we discuss two attractive applications of CX-based compressed representation, namely compressed sensing and compressed prediction.

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 $\mathbf{A}$ for any given $\mathbf{C}$. Although we still need data from all the links to precompute $\mathbf{X}_h$, this operation can easily be performed offline. Hence, during online operations, the system would only require data from a small number of sensors. We refer to this scenario as compressed sensing. It is noteworthy that low-dimensional PCA models can not be used for compressed sensing since PCA requires data from all sensors for both offline and online operations.

The underlying assumption of the proposed method is that the traffic conditions are stationary, so that a fixed matrix $\mathbf{X}$ allows us to accurately reconstruct the original data matrix $\mathbf{A}$ from $\mathbf{C}$ [3], [4]. Therefore, we can estimate the present network state ($\hat{\alpha}_p^i$) as $\hat{\alpha}_p^i = \mathbf{c}_p^i \mathbf{X}_h$ $\forall i = k, ...m$, where $\hat{\alpha}_p^i \in \mathbb{R}^{1\times n}$ is a row vector which represents the current state of the entire network for test data ($i = k, ...m$). Row vector $\mathbf{c}_p^i \in \mathbb{R}^{1\times c}$ 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 as $(\frac{n}{c})$.

Large traffic networks contain a 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 to infer the traffic state of the entire network. Although different choices of temporal and/or spatial clustering can be applied, we 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 $\mathbf{A}_h$. The issue of non-stationarity is due to the fact that matrix $\mathbf{X}_h$ is inferred from training (historical) data instead of the current data. The matrix $\mathbf{A}_p$ ($\mathbf{A}_p = [\alpha_p^k ... \alpha_p^m]^T$) is not available, and the goal is to infer that matrix by extrapolating the matrix $\mathbf{C}_p$ ($\mathbf{C}_p = [\mathbf{c}_p^k ... \mathbf{c}_p^m]^T$) 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 $\hat{\mathbf{A}}_p$ ($\hat{\mathbf{A}}_p = [\hat{\alpha}_p^k ... \hat{\alpha}_p^m]^T$) 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:

$$\frac{1}{rn}\|\mathbf{A}_p - \hat{\mathbf{A}}_p\|_F^2 = \frac{1}{rn}\|(\mathbf{A}_p - \mathbf{C}_p\mathbf{X}_p) - (\mathbf{C}_p\mathbf{X}_h - \mathbf{C}_p\mathbf{X}_p)\|_F^2, \tag{3}$$

$$= \frac{1}{rn}\|(\mathbf{A}_p - \mathbf{B}) - (\hat{\mathbf{A}}_p - \mathbf{B})\|_F^2, \tag{4}$$

$$= \frac{1}{rn}\Big(\sum_{i=k}^{m}\sum_{j=1}^{n}(a_{ij} - b_{ij})^2 + \sum_{i=k}^{m}\sum_{j=1}^{n}(\hat{a}_{ij} - b_{ij})^2 - 2\sum_{i=k}^{m}\sum_{j=1}^{n}(a_{ij} - b_{ij})(\hat{a}_{ij} - b_{ij})\Big), \tag{5}$$

where $r = (m - k + 1)$ represents the number of time instances in test data matrix $\mathbf{A}_p$. 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 (see (5)). The third component of the error refers to the correlations between aforementioned error components (see (5)). To make this interpretation more explicit, we rewrite (5) as:

$$\mathrm{MSE}_{\mathrm{est}} = \mathrm{MSE}_{\mathrm{com}} + \mathrm{MSE}_{\mathrm{ns}} - 2\xi_{\mathrm{est}}, \tag{6}$$

where ξ_{est} is correlation coefficient between compressibility and non-stationarity. We will analyze the behavior of these errors for different compression ratios in Section VI.

C. Compressed prediction

In the previous section, we inferred the condition of the entire traffic network by observing traffic conditions 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 the predicted state of a small subset of links. We recall that low-dimensional models generated by PCA can not be utilized for this task since PCA requires information for all links in the network. Instead, we use state-of-the-art algorithm to predict the traffic speed only for a selected subset of locations. Then, we utilize the proposed method to extrapolate the predictions to the rest of the network using the precomputed relationship matrix. This can be written as $\hat{\alpha}_f^i = \hat{\mathbf{c}}_f^i \mathbf{X}_h$, $\forall i = k, ...m$, where $\hat{\mathbf{c}}_f^i \in \mathbb{R}^{1\times c}$ is the row vector containing the predicted values of the traffic variable at the selected locations and i^{th} time instance, $\hat{\alpha}_f^i \in \mathbb{R}^{1\times n}$ contains the predictions for all locations at i^{th} time instance, and $\mathbf{X}_h$ is the relationship matrix. If the predictions $\hat{\mathbf{c}}_f^i$ would be identical to the true values $\mathbf{c}_f^i$, then the problem boils down to compressed sensing, which we discussed in the previous section. In practice, however, the predictions have some inaccuracies. Therefore, we can write $\mathbf{c}_f^i = \hat{\mathbf{c}}_f^i + \Delta\mathbf{c}^i$ where $\Delta\mathbf{c}^i$ represents the prediction error for the subnetwork at time i. 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 $\mathbf{C}_f$ ($\mathbf{C}_f = [\mathbf{c}_f^k ... \mathbf{c}_f^m]^T$). Then, the MSE between predicted $\hat{\mathbf{A}}_f$ ($\hat{\mathbf{A}}_f = [\hat{\alpha}_f^k ... \hat{\alpha}_f^m]^T$) and true future values $\mathbf{A}_f$ ($\mathbf{A}_f = [\alpha_f^k ... \alpha_f^m]^T$) can be written as:

$$\frac{1}{rn}\|\mathbf{A}_f - \hat{\mathbf{A}}_f\|_F^2 = \frac{1}{rn}\|(\mathbf{A}_f - \mathbf{C}_f\mathbf{X}_h) - (\hat{\mathbf{C}}_f\mathbf{X}_h - \mathbf{C}_f\mathbf{X}_h)\|_F^2, \tag{7}$$

$$= \frac{1}{rn}\|(\mathbf{A}_f - \mathbf{D}) - (\hat{\mathbf{A}}_f - \mathbf{D})\|_F^2, \tag{8}$$

$$= \frac{1}{rn}\Big(\sum_{i=k}^{m}\sum_{j=1}^{n}(a_{ij} - d_{ij})^2 + \sum_{i=k}^{m}\sum_{j=1}^{n}(\hat{a}_{ij} - d_{ij})^2 - 2\sum_{i=k}^{m}\sum_{j=1}^{n}(a_{ij} - d_{ij})(\hat{a}_{ij} - d_{ij})\Big), \tag{9}$$

where the first component of the error corresponds to the non-stationarity of spatial patterns within the network and the

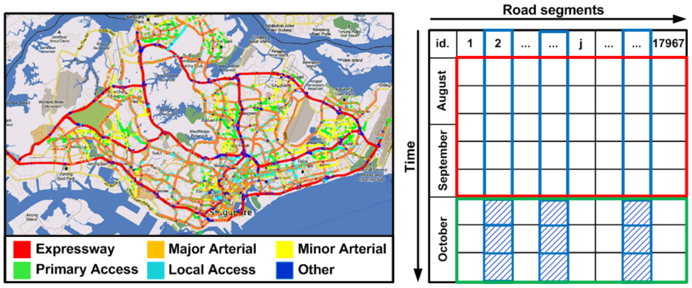
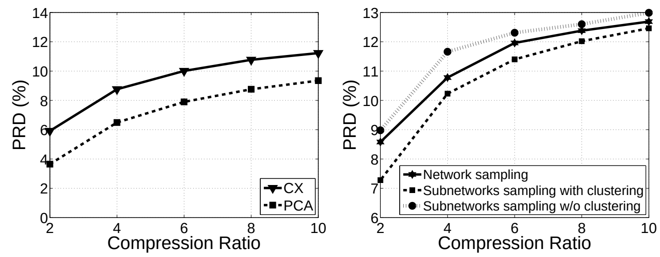


Fig. 1: *Left*: City-scale network of Singapore with 17,967 road segments of different categories, from freeways to local feeders. *Right*: Corresponding input data matrix.



(a) Compression (b) Compressed sensing

Fig. 2: Performance of the proposed approach for different applications.

second component is due to inaccurate predictions. The third component of the error shows the correlations between these two error components. We refer to (9) as MSE for compressed prediction. We rewrite (9) in more explicit form:

$$\text{MSE}_{\text{total}} = \text{MSE}_{\text{est}} + \text{MSE}_{\text{pred}} - 2\xi_{\text{pred}}, \quad (10)$$

where ξ_{pred} refers to correlation between non-stationarity and predictability. By substituting (6) in (10), we obtain:

$$\text{MSE}_{\text{total}} = \text{MSE}_{\text{com}} + \text{MSE}_{\text{ns}} + \text{MSE}_{\text{pred}} - 2\xi_{\text{est}} - 2\xi_{\text{pred}}. \quad (11)$$

Hence, 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.

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 conditions 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 algorithms, and is proportional to the number of locations c in the subnetwork. The second step (extrapolation) requires a single matrix-vector multiplication with complexity $O(\text{cn})$. 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.

V. Experimental setup

We consider the nationwide traffic network in Singapore which contains diverse types of roads (see Fig.1). The variable of interest is the average traffic speed, i.e., the average speed of all vehicles which traverse a link during the given sampling interval of 5 minutes. The data set contains the average speed at each link of the transportation network for a period of three months (August - October 2011). We selected 17967 links which had less than 5% of missing values. We performed imputation by applying the Low Dimensional CP Weighted OPTimization (LDCP-WOPT) imputation method as it is able to deal with the large data set [20], [21].

We represent the data set in the form of a matrix as explained in Section IV. For compressed sensing and compressed prediction, we need training data to: (i) determine the subnetwork of c links, corresponding to the matrix $\mathbf{C}$ (see blue rectangle in Fig. 1) ; (ii) learn relationships between the subnetwork, as defined in (i), and the entire network; (iii) train the predictors. To this end, we use the speed data of the months August and September, 2011 (see red rectangle in Fig. 1). The remaining data is used to evaluate the performance of compressed sensing and compressed prediction (see green rectangle in Fig. 1). For compressed prediction, we predict traffic variable at specific locations using a baseline predictor (see blue shaded squares in Fig. 1). We apply SVR (support vector regression) for prediction, since it is commonly used [8], [11], [12], [22]. In the baseline case, we apply SVR to each link individually. We refer to this case as uncompressed prediction. In compressed prediction, we apply SVR only to a subnetwork and next extrapolate the predictions to the entire network.

We use percent root mean distortion (PRD) error to evaluate the reconstruction error of low-dimensional models. The percent root mean distortion (PRD) quantifies the reconstruction error:

$$\text{PRD}(\%) = \frac{\| \mathbf{A} - \mathbf{CX} \|_F}{\| \mathbf{A} \|_F}. \quad (12)$$

We use the mean squared error (MSE) to assess the impact of several error components on the overall error. MSE is defined as:

$$\text{MSE} = \frac{1}{mn} \|\mathbf{A} - \mathbf{CX}\|_F^2. \quad (13)$$

VI. Results

First we investigate the CX-based method for the compression of traffic data. We apply SVD sampling strategy to find the appropriate set of columns. We repeat sampling five times and report the average reconstruction accuracy. Our benchmark is PCA, as it is considered as the optimal linear transformation. Fig. 2a shows the compression performance of the proposed and baseline methods. As expected, PCA outperforms the proposed method in terms of compression error. However, as we pointed out earlier, the low-dimensional model obtained by PCA cannot be applied for compressed estimation and prediction, since it requires data from all links. By contrast, the low-dimensional models generated by the CX method are perfectly suitable for both applications, which is the main advantage of our approach compared to PCA.

An important question about the proposed CX-based method is whether the 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

	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.

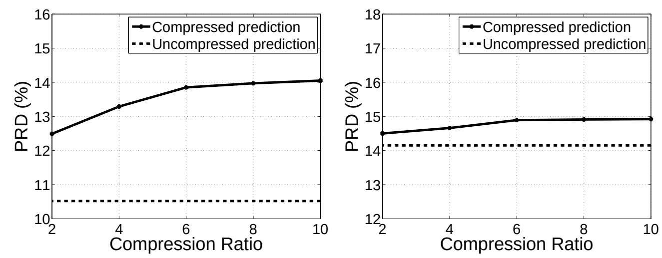
each month, we sort the road segments in descending order according to the assigned probability by the SVD sampling method. Hence, the most representative roads are at the top of these lists. Next we select the first k links of each list, with k corresponding to 10%, 25%, and 50% of the links in the network. If the subnetwork is stable across time, the three short lists of top-k 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 it 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.

We now investigate the case of compressed sensing. We aim to reconstruct the average speed at each link in the entire network by collecting data from a small subset of roads. The relationship matrix $\mathbf{X}_h$ is determined from the training set (data from Aug-Sep, 2011), and the reconstruction error is assessed on the test set (data from October, 2011). Fig. 2b shows the reconstruction accuracy of the proposed method for three different approaches: In the first approach we select the subset of road segments according to SVD sampling scheme (see solid line in Fig. 2b). In the second approach 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. 2b). In the third approach we use the identical set of roads as defined in the second approach to perform network estimation. Unlike in the second approach, we do not perform any clustering here (see dotted line in Fig. 2b). Intuitively, the difference between the second and the third approach shows the gain obtained by network clustering. Fig. 2b 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. Let us now investigate the error of compressed sensing in more details. In our analysis, we consider the subset of links as defined in SVD scheme without clustering.

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 MSE of the individual error components, the correlation between the two errors components and the total MSE, for different compression ratios. As it can be seen from Table II 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. Furthermore,

	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.



(a) 5 min prediction horizon (b) 30 min prediction horizon

Fig. 3: Prediction performance of the proposed and traditional methods.

Table II also shows that there is no correlation between compressibility and non-stationarity of traffic data.

The third application of the proposed CX-based 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 (baseline) approach where the speed for each road segment is explicitly predicted. Fig. 3 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 error than the traditional approach. This additional error decreases with decreasing compression ratio. Also, the additional error decreases for large prediction horizons (see Fig. 3b). Naturally, it is difficult to predict for larger horizons even with traditional approach (see dashed line in Fig. 3b). Hence, in such cases the error due to prediction tends to become the dominant component.

We decompose the 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 for 5 minute prediction horizon. As it 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 of the error component (see Table II). Table III shows that the prediction error tends to be dominant for smaller compression ratio, i.e., when significant portion of the network is explicitly predicted. From Table III, we can also see that there is some correlation between the two error components.

The proposed approach of compressed prediction provides substantial reduction in computational complexity by explicitly predicting the variables at a small representative set, followed

	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.

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: Computation time (in seconds) of the compressed method for 5 minute prediction horizon. The traditional approach requires 91.63 sec to perform prediction for the whole network.

by (fast and efficient) extrapolation to the entire network. This reduction in computation time is obtained at the expense of a small increase in the prediction error (see Fig 3). The computation 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 2.67 GHz MacPro server on a single core with 32GB of random-access memory (RAM). We assume that training phases are performed offline for both methods. Prediction time for compressed method involves the time to predict traffic variable for a subset of links and time required to perform network wide extrapolation. As Table IV shows, the latter can be neglected. Consequently, the required computation time for compressed prediction is proportional to the number of the road segments in **C**. Consequently, the reduction in computational complexity is approximately proportional to the compression ratio (see Table IV).

VII. Conclusions

In this paper we utilized column based (CX) low-dimensional models to enhance the scalability of compressed sensing and compressed prediction. We decomposed the compressed prediction error into several components and investigated the relationship between them. Our numerical results show that the proposed method significantly reduces the computational cost at the expense of a negligible increase in prediction error.

In future work, we will explore whether other column selection techniques can lead to better performance of compressed prediction. Also, we will investigate how compressed prediction can be applied in conjunction with routing, in order to optimize routes taking future traffic conditions into account.

VIII. Acknowledgment

Justin Dauwels wishes to thank Andrzej Cichocki for helpful discussions.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

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

[12] 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.

[13] 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.

[14] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Subspace sampling and relative-error matrix approximation: Column-based methods," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2006, pp. 316–326.

[15] C. Boutsidis, M. W. Mahoney, and P. Drineas, "Unsupervised feature selection for principal components analysis," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 61–69.

[16] 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.

[17] 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.

[18] 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

[19] 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.

[20] M. T. 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.

[21] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

[22] E. J. Horvitz, J. Apacible, R. Sarin, and L. Liao, "Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service," *arXiv preprint arXiv:1207.1352*, 2012.