Compressive Sensing Approach to Urban Traffic Sensing

Zhi Li[†], Yanmin Zhu[†*], Hongzi Zhu[‡] and Minglu Li[†]
[†] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[†] Shanghai Key Lab of Scalable Computing and Systems; [‡] University of Waterloo
[*] Corresponding Author
[†]{lzhi, yzhu, mlli}@sjtu.edu.cn, [‡]hongzi@bbcr.uwaterloo.ca

Abstract*—Traffic sensing is crucial to a number of tasks such as traffic management and city road network engineering. We build a traffic sensing system with probe vehicles for metropolitan scale traffic sensing. Each probe vehicle senses its instant speed and position periodically and sensory data of probe vehicles can be aggregated for traffic sensing. However, there is a critical issue that the sensory data contain spatiotemporal vacancies with no reports. This is a result of the naturally uneven distribution of probe vehicles in both spatial and temporal dimensions since they move at their own wills. This paper proposes a new approach based on compressive sensing to large-scale traffic sensing in urban areas. We mine the extensive real trace datasets of taxies in an urban environment with principal component analysis and reveal the existence of hidden structures with sensory traffic data that underpins the compressive sensing approach. By exploiting the hidden structures, an efficient algorithm is proposed for finding the best estimate traffic condition matrix by minimizing the rank of the estimate matrix. With extensive trace-driven experiments, we demonstrate that the proposed algorithm outperforms a number of alternative algorithms. Surprisingly, we show that our algorithm can achieve an estimation error of as low as 20% even when more than 80% of sensory data are not present

Keywords-traffic sensing, probe vehicles, compressive sensing, traffic matrices, interpolation

I. INTRODUCTION

Traffic sensing aims to detect the traffic conditions of different roads. It is crucial to a number of tasks, such as traffic management, road engineering and infrastructure planning. Shanghai, for example, the largest metropolis in China, is undergoing rapid economic growth, but meanwhile suffers constant traffic congestion. To mitigate the burden of the underlying road networks, efficient traffic management based on metropolitan scale traffic sensing is essential. Traffic sensing, however, is a great challenge in a large scale metropolis like Shanghai, China.

Traditionally, vehicle loop detectors and close-circuit cameras are deployed at roadside to detect passing vehicles [3, 6]. Unfortunately, the coverage of these systems is supremely limited due to the high deployment and maintenance costs. For example, a loop sensor costs $900-$2000 dependent on its type. More importantly, deployment, direct and indirect maintenance costs are significant. It is practically infeasible to install traffic monitoring systems densely enough to cover the entire road networks. Researchers and engineers are trying to overcome the limitations of loop detectors and cameras by using new technologies, for example, RFID. Some studies try to locate vehicles based on cellular phone base station [1]. But these approaches also encounter the coverage problem.

We build a traffic sensing system with probe vehicles which are voluntary and public vehicles such as taxies. This system constitutes a novel way to traffic sensing based on GPS-equipped vehicles and wireless communication. Specifically, vehicles are utilized as mobile sensors for real-time traffic information collection. In this system, more than 4000 taxies are employed, each periodically sensing its instantaneous speed and position. The sensory data about traffic is reported to a central data center via a cellular wireless channel. By aggregating these traffic sensory data, metropolitan scale traffic sensing is possible. In Fig. 1, a snapshot of vehicle distribution over the Shanghai downtown region is shown. This approach to traffic sensing has salient advantages. First, as these public vehicles traverse most of the road segments in the city, the approach has a supremely large coverage. Second, because of the cheap price of onboard GPS devices and wireless communications, the overall system cost is low. Third, the accuracy of traffic sensing can be improved by leveraging more probe vehicles.

However, it is still challenging to get traffic condition information directly from the raw reports. First, the report data usually contain a lot of vacancies in both time and space. The amount of vacancies is dependent on the number of probe vehicles and their movement trajectories. The distribution of probe vehicles over space and time is uneven. It is apparent for a popular downtown region to have more probe vehicles moving around. It is also intuitive that rush hours have more probe vehicles than normal hours. Second, individual traffic reports are error-prone. Under ideal conditions, the GPS po-

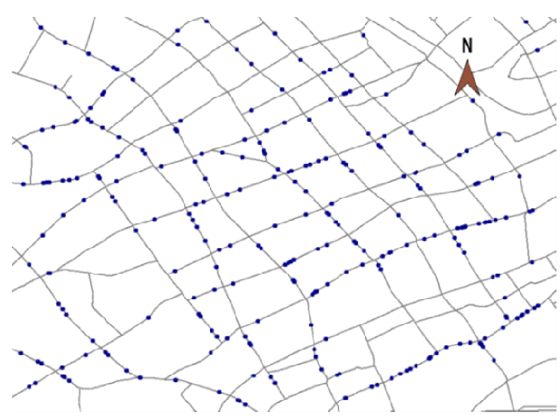


Figure 1. A distribution snapshot of probe vehicles on road network in Shanghai downtown region (a dot is a probe taxi).

1063-6927/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCS.2011.35

sition can be highly accurate, but when the probe vehicle passes through many obstructions including so-called urban canyons and underpasses that degrade or impede operation. Multipath propagation of radio signals also creates problems. Some study [9] focuses on how to detect a GPS error of a probe vehicle. In addition, unreliable wireless communication may introduce noises and errors.

Some existing work has already started to look into details at how we can process traffic data based on these systems. In [8], the authors discuss the architecture of probe vehicle system and develop a simple analytical/statistical model. They figure out that 3% penetration is required on highways and over 5% penetrate is required on surface roads. Some work [14] focuses on figuring out the street traffic state on a given road segment based on trace data of probe vehicles. Abundant data are required in their measure and each road segment is analyzed independently. Virtual Trip Lines are approached in [10] to protect the privacy of individual trace. The existing studies do not solve the problem of vacancies in sensory traffic data. In [16], the authors try to reconstruct missing values by using Multiple Singular Spectrum Analysis (MSSA). However, it can only exploit the internal periodicities in the sensory data and there is still much space for improvement.

In this paper, we propose a new approach based on compressive sensing to large-scale traffic sensing. Compressive sensing [2] is a powerful theoretical tool for dealing with missing values in real-world datasets that usually contain certain structures. It has recently attracted considerable attention in statistics, signal processing and approximation theory. According to compressive sensing, a real-world dataset matrix can well be approximated by a low-rank matrix. For locations and times of interest, we define a traffic condition matrix. We mine the large traces of more than 4000 taxies over duration of more than two years in Shanghai, China by using Principal Component Analysis (PCA). We find that there evidently exist hidden structures with traffic condition matrices. By applying singular value decomposition, the major energy concentrates on just a few principle components. This shows that the datasets contain hidden structures or redundancy which underpins the applicability of compressive sensing to traffic sensing.

Based on the observations, we propose a compressive sensing based algorithm to exploit hidden structures for traffic sensing. The algorithm tries to find the optimal estimate matrix for the original matrix. According to the principle of compressive sensing, the objective for finding the optimal estimate is achieved by minimizing the rank of the estimate matrix. Meanwhile, the estimate matrix should be as close as possible to the measurement matrix. We propose an efficient genetic algorithm for solving the optimization problem and find the best estimate. Experiments based on large real trace datasets demonstrate that our algorithm produces the best performance, outperforming alternative algorithms, such as K-Nearest Neighbors (KNN) and Multi-channel Singular Spectrum Analysis (MSSA). Surprisingly, we show that our algorithm can achieve an estimation error of as low as 20% even when more than 80% of sensory data are not present.

In this paper, we have made the following contributions.

- We mine the large traces of taxies in Shanghai with principle component analysis and reveal the hidden structures in traffic condition matrices.
- We propose a compressive sensing based approach to finding the best estimate for original traffic condition matrices. Design optimizations of the algorithm are accomplished by a genetic-based optimization algorithm.
- We conduct real trace driven experiments. Performance results demonstrate that our algorithm achieves significantly lower estimation error than alternative algorithms.

The rest of this paper is organized as follows. In Section II, we formally formulate the problem. In Section III, we mine the trace datasets and reveal the existence of hidden structures in the datasets. The compressive sensing based algorithm is detailed in Section IV. In Section V, we present our experiment results. We introduce related work in Section VI. We conclude the paper in Section VII.

II. Problem Formulation

We give the system model and formally formulate the problem in this section.

A. System Model

Probe mobile vehicles are deployed for sensing traffic information of roads. There are N vehicles in the system, denoted by a set of $\{0, 1, \dots, N-1\}$. For a probe vehicle, i, it moves along the roads and reports its states from time to time. The reported states at time t are a three-tuple, $s_i(t): < v, a, p >$, representing its instant velocity, direction of head way, position (longitude & latitude). A sensory report is delivered back to a central server by a wireless data channel. Note that the sensory data itself may contain error and a sensor report is not guaranteed to be delivered to the server. Importantly, a vehicle travels at its own will and there is no central scheduling for movement of vehicles.

Let T_i denote the set of timestamps at which vehicle i reports its states, $T_i = \{t_1^i, t_2^i, \dots, t_k^i,\}$, in which t_1^i and t_k^i are the first timestamp and the last timestamp, respectively. Thus, vehicle i forms a report set, $S_i = \{s_i(t) \mid t \in T_i\}$. Note that for different vehicles, the set of timestamps may be different. For a vehicle, the intervals between timestamps are not necessarily the same.

B. Problem Statement

It is not straightforward to devise a single metric for quantifying the traffic condition of a given location at a given time. In general, a good traffic condition allows higher driving speed and larger throughput. This paper adopts the driving speed as the metric for quantifying traffic condition.

Definition 1 (traffic condition): Given a target location on a road, p_0 (px_0, py_0), the traffic condition of this location at time t_0 is defined as the driving speed of p_0 at time t_0, denoted by $x(p_0, t_0)$.

Since it is impossible to get $x(p_0, t_0)$ directly, we can only estimate this value by using the driving speeds of vehicles passing by this location. A good estimate is the mean of these driving speeds.

It is challenging, however, to obtain the average of the driving speeds. The number of available reports whose locations and timestamps exactly match (p_0, t_0) is negligible. A reasonable way to circumvent this obstacle is to use reports which reside in the neighborhood of (p_0, t_0). Let $R(p_0, t_0)$ denotes the set of reports in the neighborhood

$$R\left(p_0, t_0\right) = \left\{S_i\left(t\right) \mid d\left(p, p_0\right) \le \Delta p \,\&\&\, \left|t - t_0\right| < \Delta t\right\} \quad (1)$$

Then, the estimate is computed as

$$\hat{x}\left(p_0, t_0\right) = \frac{1}{K} \sum_{S_i(t) \in R(p_0, t_0)} S_i\left(t\right).v \quad (2)$$

where $K = |R(p_0, t_0)|$.

It is obvious that the quality of this estimate depends on the three factors, i.e., K, Δp and Δt. Usually the two values, Δp and Δt are selected by the users and remain fixed for a given application scenario. This paper relaxes the spatial granularity Δp. We consider the traffic condition on a road segment, which refers to the part between two neighboring intersections of a road in one direction.

The number of reports, K, is usually a random value, influenced by the total number of probe vehicles, the location and the traffic condition. As a result, K impacts the quality of estimatimation. Under a good condition, K is large and the estimate would be close to the real value according the central limit theory. Under the worst case, K is zero, i.e., there are no reports in the neighborhood. In this case, there is no way to get an estimate.

We are interested in the traffic conditions of a given set of locations Ω at a given set of time slots Z,

$$\Omega = \left\{p_0, p_1, p_2, \ldots p_{n-1}\right\}. \quad (3)$$

$$Z = \left\{t_0, t_1, t_2, \ldots t_{m-1}\right\}. \quad (4)$$

The traffic condtions of Ω at Z form a traffic condition matrix (TCM), denoted by X_{TCM},

$$X_{TCM} = \left(x_{ij}\right)_{m \times n}. \quad (5)$$

where x_{ij} is the average driving speed of $R(p_j, t_i)$. Therefore, a row gives traffic conditions of different locations at t_i and a column specifies a time series of traffic condition at the location p_j.

It is difficult to obtain an integral traffic condition matrix as there may be a lot of vacancies with no reports. Probe vehicles move on roads according with their own preferences, and there is no guarantee for a location that a number of reports are generated for every given time period. This creates the problem of report holes that there are no sufficient reports for computing traffic conditions. This will further be demonstrated in the following section.

In fact, we are given a measurement matrix (M_{TCM}):

$$M_{TCM} = X_{TCM} . \times B$$
$$B = [b_{ij}] = \begin{cases} 0, & \text{if } |R(p_0, t_0)| = 0 \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

where matrix B is an indication matrix. The objective is to estimate X_{TCM} when given M_{TCM}. Formally, our traffic sensing problem is defined as follows:

Definition 2 (Traffic sensing problem). Given the probe sensors and their reports, the traffic sensing problem is to find an estimate, $\hat{X}_{TCM}$, such that it approximates the original traffic matrix, X_{TCM}, as closely as possible.

Note that when the time set of interest includes future times, by solving the traffic sensing problem we are able to predict future traffic conditions.

III. Real-World Data Analysis

We have deployed more than 4000 taxies as probe vehicles in Shanghai, China, for large scale traffic sensing. Each vehicle reports its sensory data periodically. The period varies from vehicle to vehicle between ten seconds and three minutes. We have collected a large amount of traces of the taxies. The trace datasets span duration of more than two years. We first demonstrate the serious issue of report holes by studying the distribution of sensory reports over both space and time. Next, we reveal that there exist certain structures in the sensory data by applying principal component analysis.

A. Distribution over Time and Space

We study the distributions of sensory reports over space and time, by which we show the issue of report vacancies. We define a metric as follows.

Definition 3: Let B be the indication matrix for matrix M. The integrity of M, denoted by $\varpi(M)$, is defined as the ratio of available elements to the total elements.

$$\varpi(M) = \text{sum}(B) \,/\, \text{size}(B). \quad (7)$$

We analyze the traces of 500, 1000, 2000 taxies over a duration of 24 hours on Feb 18, 2007, respectively. All the taxies were running in the inner region of Shanghai, in which there are 5812 road segments. We set the time granularity to 15 minutes in the analysis.

First, we study the integrity of measurements at a given road, by which we can learn the issue of report holes of sensory data over time. Fig. 2 shows the CDFs of integrity of all roads under different numbers of vehicles, i.e., 500, 1000 and 2000. We can see that when there are 500 probe vehicles, nearly 95% of roads have an integrity of less than 60%. This means that these roads have no reports for more than 40% of time. Generally, when we deploy more probe vehicles, the integrity can be improved. However, even when 2000 probe vehicles are employed, there are still nearly 80% of roads whose integrity is less than 60%. More importantly, we find that nearly 50% of the roads have an integrity close to zero. This suggests that not a single vehicle passes by these roads.

Next, we consider the integrity of measurements at a given time snapshot. In this way, we can learn the issue of report holes of sensory data over space. In Fig. 3, we plot the CDFs of integrity of all time slots under different numbers of probe vehicles, i.e., 500, 1000 and 2000. We can see that when there are 500 probe vehicles, nearly 100% of time slots have an integrity of less than 18%. This indicates that almost for all times, more than 82% of roads have no reports.

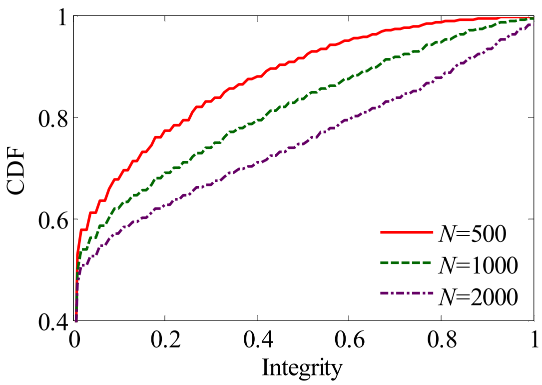


Figure 2. CDF of integrity of roads.

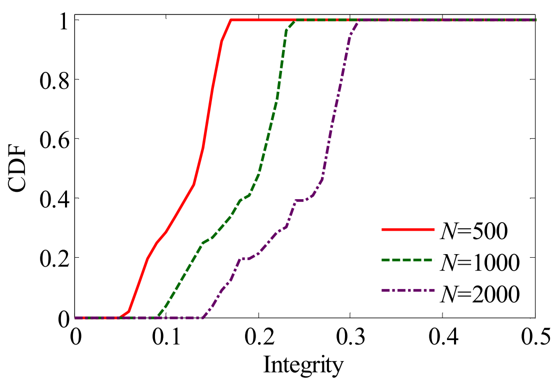


Figure 3. CDF of integrity of time slots.

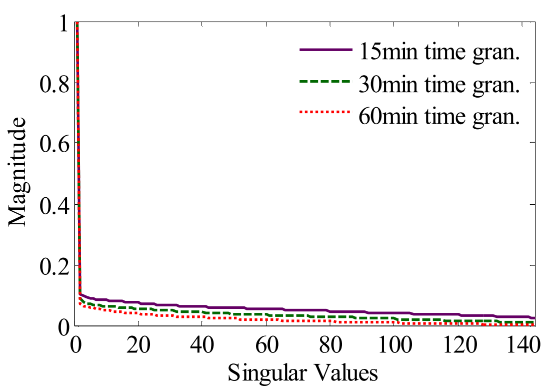


Figure 4. Magnitude of singular values.

TABLE I. INTEGRITY SUMMARY (FEB 18, 2007)

Time gran.	***N*=500**	***N*=1000**	***N*=2000**
15 min	12.22%	18.28%	24.80%
30 min	18.57%	25.18%	31.61%
60 min	25.53%	31.98%	37.64%

Finally, we study the integrity for different time granularities. Table I shows the integrities under different time granularities when there are 500, 1000 and 2000 probe vehicles. We can find that even when there are 2000 probe vehicles, the integrity is as low as 24.8% when time granularity is 15 minutes and 37.64% when time granularity is 60 minutes.

In summary, the issue of report holes is serious. The possible solution to improving integrity is by deploying more probe vehicles. However, this may increase cost, and it may be impractical in some situations, e.g., there is no way to employ more probe vehicles.

B. *Structure Discovery in Sensory Data*

The traffic conditions of different locations over different times are not independent. There exist structures. We reveal such hidden structures by using principal component analysis (PCA), which is an effective non-parametric technique for revealing sometimes hidden, simplified structure that often underlie a dataset [11].

Any matrix X can be decomposed into three matrices according to singular value decomposition (SVD):

$$X = USV^T = \sum_{i=1}^{\min(n,m)} \sigma_i u_i v_i^T \tag{8}$$

where V^T is the transpose of V, U is a $n \times n$ unitary matrix (i.e., $U^T U = UU^T = I_{n\times n}$), V is a $m \times m$ unitary matrix (i.e., $V^T V = VV^T = I_{m\times m}$), and S is a $n \times m$ diagonal matrix constraining the singular values σ_i of X. Let σ_i be larger than σ_{i+1}, $i = 1,2,..,l$, where l is the rank of X. The rank of a matrix equals the number of its non-zero singular values. Here v_i is the unit eigenvector of $X^T X$ corresponding to the i-th principal component. We call u_i an eigenflow of X [11].

$$u_i = \left(Xv_i\right)/\sigma_i\,,\ \text{i}=1,2,..,\min(m,n)\,. \tag{9}$$

According to (8), σ_i is a coefficient of the i-th principal component which we may explain as the energy of the i-th principal component.

In Fig. 4, we present the magnitude (ratio to the maximum) of singular values. This figure suggests that most of the energy is contributed by the first few principal components. The existence of the sharp knee is a result of some common structures among different interested locations, which will lead the traffic condition matrix to a low rank.

The information of a dataset is mainly contained by the first few components. We reconstruct the traffic matrix using the only first five principal components. Fig. 5 shows the reconstructed traffic condition over times of a given location in which the time granularity is 30 minutes. It can be observed that the reconstructed traffic conditions well sketches the variation of the original ones.

Then we look into the characteristics of eigenflow u_i. A time series X_i can be presented as a line combination of u_i with associated weight $(V^T)_i$.

$$X_i = \sigma_i U(V^T)_i, \qquad i = 1,2,..,\min(m,n) \tag{10}$$

where $(V^T)_i$ is the i-th row of V.

All the eigenflows can be divided into three types. Let $C(u_i) \in \{1,2,3\}$ denote the type of an eigenflow, $v_i, 0 \le i < \min(m,n)$. Its type is determined as follows,

$$C(u_i) = \begin{cases} 1, & \text{if } |FFT(u_i)| \text{ contains a spike} \\ 2, & \text{if } u_i \text{ contains a spike} \\ 3, & \text{otherwise} \end{cases} \quad . \tag{11}$$

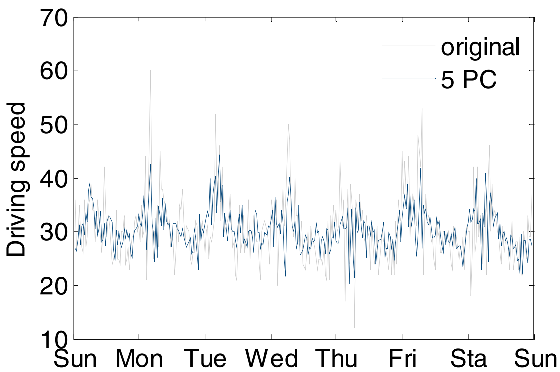


Figure 5. Original and reconstructed traffic conditions of a given location using first 5 principal components (gran. is 30 minutes).

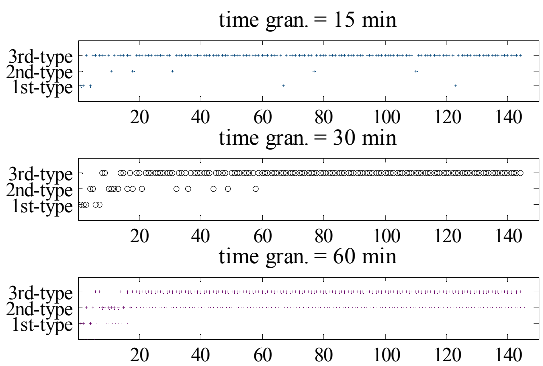


Figure 6. Occurrence of eigenflow types in the corresponding order of singular values.

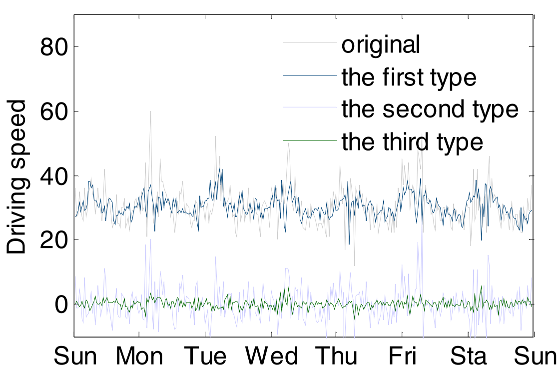


Figure 7. Reconstructed traffic conditions of a given location by using different types of eigenflow.

If the difference of the value and the average is larger than four times the standard deviation, the value is a spike.

When the signal of an eigenflow is periodic (i.e., its FFT energy contains an evident spike), this eigenflow belongs to the first type. An eigenflow of the first type is considered as deterministic, i.e., this type of eigenflows contains the majority of information in the datasets. If an eigenflow does not belong to the first type and its signal contains a spike, it belongs to the second type. The spike in the eigenflow of the second type indicates that the original datasets also have a corresponding spike. The rest of the eigenflows belong to the third type. An eigenflow of the third type contains little information and can be considered as containing only noises. This explanation is illustrated in Fig. 8.

Fig. 6 shows the occurrences of eigenflow type in the order of singular values. The most important information often comes from the eigenflows of first type, which correspond to singular values. In Fig. 7, we reconstruct the traffic conditions over time at a given location by using different types of eigenflows. We find that the first type contains most information and well sketches the variation of the original series of traffic conditions. The second and the third types contain little information with a mean value close to zero.

In summary, the results based on principal component analysis demonstrate that there are hidden structures with traffic condition matrices. This lays the foundation for our compressive sensing based approach for traffic sensing.

IV. Compressive Sensing Based Algorithm

The objective of traffic sensing is to compute an estimate of traffic condition matrix that approximates the real traffic condition matrix as closely as possible. There are many ways to compute an estimate, but no obvious way can compute the best estimate. We propose a compressive sensing based approach which effectively exploits the hidden structures associated with sensory data.

A. Compressive Sensing

We have revealed that there exist hidden structures in traffic condition matrices. Compressive sensing is an effective technique for a number of tasks, such as data compression and signal processing [5, 7]. The main idea of compressive sensing is that signals or datasets in the real world often contain structures or redundancy (i.e., they are not pure random noises). This nature can be used as prior knowledge for compression and reconstruction of signals or datasets.

Mathematically, a vector with only a few non-zero elements is called a sparse vector. Structure or redundancy in datasets is synonymous with sparsity. A matrix of dataset may have only a few large elements and many small elements. Such a vector is considered as compressible, in the sense that most of its information is actually carried in the large elements. A sparse matrix can be well approximated by a low rank matrix.

As shown in Section III, any matrix can be decomposed in such a way that it equals the multiplication of three component matrices. When the rank is fixed and set to r, to generate an estimate that approximates the original matrix, we keep the r largest components in (8) and drop the others. Thus,

$$\hat{X} = \sum_{i=1}^{r} \sigma_i u_i v_i = \sum_{i=1}^{r} \sigma_i A_i. \qquad (12)$$

This $\hat{X}$ is known as the best rank-r approximation with respect to the Frobenius norm $||\cdot||_F$ of approximation errors, $||X||_F \triangleq \sqrt{\sum_{i,j} X_{ij}^2}$ for any matrix. Then, $\hat{X}$ is the solution to the following optimization problem,

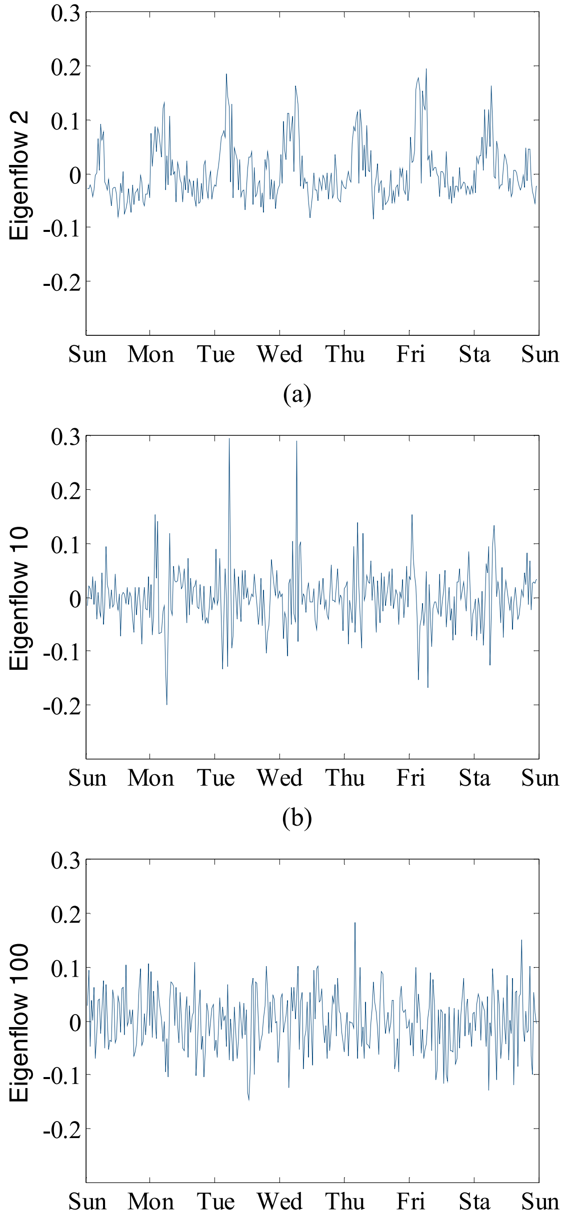


Figure 8. Time series represented by three types of eigenflows, (a): the first type; (b): the second type; (c): the third type

$$\begin{aligned} &\min \| X - \hat{X} \|_F \\ &s.t.\ \text{rank}(\hat{X}) \le r. \end{aligned} \tag{13}$$

B. Algorithm Design

To solve the traffic sensing problem, we are given the measurement matrix and required to compute the estimate for the original matrix. It is impossible to directly applying (13) for solving this problem as we do not have the knowledge of the original matrix and the proper rank.

As a good estimate, it is reasonable to be as close as to the measurement matrix. In addition, the estimate matrix should have a low rank as we have revealed in the real datasets that they contain certain structures or redundancy. Thus, we try to find the low rank estimate as follows,

$$\begin{aligned} &\min\ \text{rank}(\hat{X}) \\ &s.t.\ B.\times \hat{X} = M. \end{aligned} \tag{14}$$

It is difficult to solve this minimization problem because it is non-convex.

To circumvent the difficulty, we make use of the SVD-like factorization, which re-write (2) as follows,

$$\hat{X} = U\Sigma V^T = LR^T \tag{15}$$

where $L = U\Sigma^{1/2}$ and $R = V\Sigma^{1/2}$. According to the compressive sensing literature [4, 12-13], we can solve a simpler problem and obtain an equivalent result under a certain condition. Specifically, if the restricted isometry property holds [12], minimizing the nuclear form can perform rank minimization exactly for a matrix of low rank. That is, we just find matrix L and R that minimize the summation of their Frobenius norms:

$$\begin{aligned} &\min \| L \|_F^2 + \| R^T \|_F^2 \\ &s.t.\ B.\times (LR^T) = M. \end{aligned} \tag{16}$$

In practice, L and R that strictly satisfy the constraint are likely to fail for two reasons. First, there are noises in the sensory data, and therefore strict satisfaction may lead to the over-fit problem. Second, a traffic condition matrix can well be approximated by a low rank matrix while its real rank may not necessarily be low.

Thus, we use the Lagrange multiplier method to solve (16),

$$\begin{aligned} &\min\ (x + \lambda y), \text{where} \\ &x = \| B.\times (LR^T) - M \|_F^2 \\ &y = \| L \|_F^2 + \| R \|_F^2 . \end{aligned} \tag{17}$$

The Lagrange multiplier λ controls the tradeoff between rank minimization and measurement fitness.

Many methods can solve the above optimization problem. We propose an algorithm that is similar to the one in [15]. We show the detail Pseudo code of this algorithm in Fig. 9.

It is an efficient heuristic algorithm. With a random initialization, the algorithm solves this optimization by first fixing matrix L, and then computes the other matrix, R. Next, R is fixed and L is computed. This process repeats until the optimal value is reached. From (17), we find that reaching the objective is equivalent to making both x and y equal zero simultaneously. Thus, we have the following when L is given,

$$\begin{bmatrix} B.\times (LR^T) \\ R \end{bmatrix} = \begin{bmatrix} M \\ 0 \end{bmatrix}. \tag{18}$$

This is a contradictory equation since the number of constraints is larger than that of unknown variables. By computing the best approximate solution to this contradictory equation, we can compute the best matrix R for satisfying (17).

We analyze the complexity of the algorithm as follows. The key operation of Algorithm 1 is the procedure for computing an inverse matrix, which gives the best approximate

Algorithm 1.
Input:
$M_{m\times n}$: measurement matrix
$B_{m\times n}$: indication matrix
r: rank bound
λ: tradeoff coefficient
t: iteration times
Output:
$\hat{X}_{m\times n}$: estimate matrix

$\mathcal{L}$←random_matrix(m, r);
for k←1 **to** t **do**
  $\mathcal{R}$←inverse([$\mathcal{L}$; $\sqrt{\lambda}I$], [M; 0]);
  $\mathcal{L}$←inverse([$\mathcal{R}^{\mathrm{T}}$;$\sqrt{\lambda}I$], [M^T; 0]);
  v←$||B.\times(\mathcal{L}\mathcal{R}^{\mathrm{T}})- M||_{\mathrm{F}}^2+\lambda(||\mathcal{L}||_{\mathrm{F}}^2+||\mathcal{R}^{\mathrm{T}}||_{\mathrm{F}}^2)$;
  if $v < \hat{v}$ **then**
    $\hat{\mathcal{L}} \leftarrow \mathcal{L}$; $\hat{\mathcal{R}} \leftarrow \mathcal{R}$; $\hat{v} \leftarrow v$;
  end if;
end for
$\hat{X}$←$\hat{\mathcal{L}} \times \hat{\mathcal{R}}^{\mathrm{T}}$;
return $\hat{X}$;

// return solution to contradictory equation
procedure inverse($\boldsymbol{P}, \boldsymbol{Q}$)
  $C \leftarrow P^{\mathrm{T}}P \setminus P^{\mathrm{T}}Q$;
  return C;

Figure 9. Pseudo code of Algorithm 1.

Algorithm 2.
Input:
$\ell_r, \mathcal{U}_r$: lower bound and upper bound of r
$\ell_\lambda, \mathcal{U}_\lambda$: lower bound and upper bound of λ
B: measurement matrix
Algorithm 1
Output:
Optimal r and λ

$\mathcal{N}$(*population*) ← initialize with random numbers uniformly distributed within $[\ell_r, \mathcal{U}_r]$ and $[\ell_\lambda, \mathcal{U}_\lambda]$
while (!stall(fitness)) **do**
  $\mathcal{H}$ ← select($\mathcal{N}$)
  $\mathcal{C}$ ← crossover($\mathcal{N}$)
  $\mathcal{M}$ ← mutate($\mathcal{N}$)
  $\mathcal{N}$ ← [$\mathcal{H}, \mathcal{C}, \mathcal{M}$]
end while
$[r, \lambda]$ ← decode (the best kid in $\mathcal{N}$)

Figure 10. Pseudo code of Algorithm 2.

solution to the contradictory equation. The procedure is essentially completed by a matrix multiplication. Therefore, its complexity is $O(rmn)$ where r, m, n denote the column number of L, the row number of X, the column number of X, respectively. The algorithm repeats the procedure for t times. Therefore, the total complexity of the algorithm is $O(rmnt)$.

C. Design Optimization

Two important parameters must be determined in Algorithms 1, i.e., rank bound r and tradeoff coefficient λ. The two parameters greatly influence the final estimate quality. According to the principle of compressive sensing, the rank of the approximated matrix should be minimized. In Algorithm 1, r is the number of columns in matrix L and R, which is smaller than m and n. Thus, we have

$$\mathrm{rank}(\hat{X}) \leq \min(\mathrm{rank}(L),\ \mathrm{rank}(R)) = r \quad (19)$$

Thus, r is an upper bound of rank($\hat{X}$), and impacts the algorithm performance.

We should determine the optimal parameters in order to obtain the best estimate. However, it is not trivial to determine the optimal parameters. The quality of estimation is a function of the two parameters, denoted by, $\ell = f(r, \lambda)$. Then, to obtain the optimal parameters, the objective is the following,

$$\max \ell = \max f\left(r, \lambda\right) \quad (20)$$

We use estimation error to indicate the quality of estimation. The definition of estimation error will be given in the next subsection. The key issue is that function $f(\cdot)$ characterizing the relationship between error and the parameters is invisible.

We propose a genetic algorithm for deriving the optimal rank bound and tradeoff coefficient. The strength of this algorithm is that the analytical form of the objective is not required to be explicitly given. In this algorithm, estimation errors are used as fitness. We encode the two parameters as a vector which contains two real numbers.

We randomly initialize the population that will evolve over generations. The best individuals encoding the optimal parameters will be selected in the end. In every generation the population consists of three groups of individuals. The first group includes the kids selected from the last generation according to their fitness. We employ the roulette model. The second group consists of the kids produced by taking the crossover of any two individuals. The final group of kids is produced by the mutation operation. Specifically, we assign a random value to one of parameters within its domain to achieve the mutation. Algorithm 2 iterates until the fitness of the best individual stall. The detail pseudo code of this algorithm is shown in Fig. 10

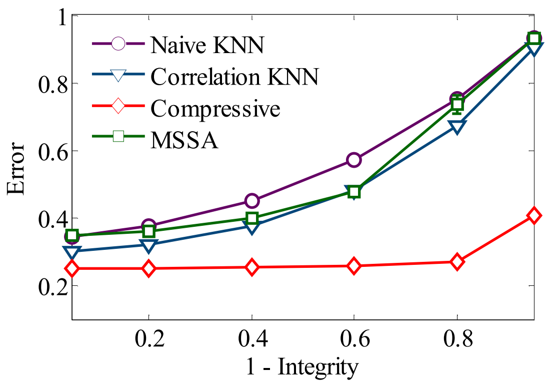
V. Performance Evaluation

We have performed extensive experiments for evaluating the performance of the proposed algorithm for traffic sensing. In the following, we first present the methodology and setting. The compared algorithms are then introduced. Finally, performance results are presented and discussed.

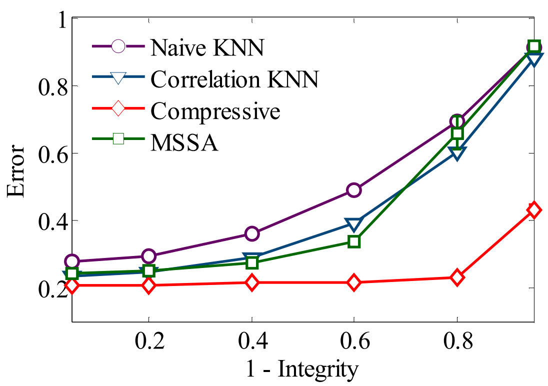
A. Methodology and Setting

We adopt a competitive study, comparing our algorithm with other alternative algorithms that will be introduced in the following subsection.

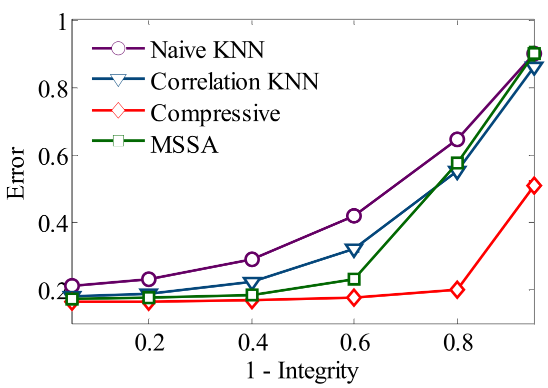
Estimation error is the performance metric for evaluation and comparison. It is defined as the normalized mean absolute error,



(a) Time gran. = 15 min



(b) Time gran. = 30 min



(c) Time gran. = 60 min

Figure 11. Performance comparisons of different algorithms (estimation error against integrity)

$$\xi = \sum_{i,j:M_{ij}=0} | X_{ij} - \hat{X}_{ij} | \Big/ \sum_{i,j:M_{ij}=0} | X_{ij} |, \tag{21}$$

where X is original matrix and $\hat{X}$ is the estimated matrix.

Experiments are conducted using the real trace datasets of more than 4000 taxies in Shanghai, China. The trace datasets span a duration of one week. Three time granularities, i.e., 15 minutes, 30 minutes and 60 minutes, are used.

We choose a downtown region for experiments. The major reason for choosing a downtown region is that we need to know the original traffic condition matrix. In reality, it is very difficult to find a fully integral matrix without vacancies. For this reason, it is better to find a matrix that is as integral as possible. When performing experiments, we randomly discard some elements to form measurement matrices. Then, these estimates are compared with the original matrices and estimation errors can be computed since the original matrices have only a few unavailable elements. Note that the calculation of estimation error does not include those elements that are unavailable in the original matrices.

B. *Compared Algorithms*

We compare our algorithm with three other algorithms.

1) *Naïve KNN*

K-Nearest Neighbors (KNN) is a simple algorithm but often used to solve many machine learning problems including recovery of missing values. The naïve KNN interpolates missing values by taking the average of its nearest K neighbors in the measurement matrix.

2) *Correlation-based KNN*

The correlation-based KNN is more sophisticated compared with the naïve one. It calculates the average by using the K neighbors from its immediate rows. The key idea is that for average computation, the candidate value is weighed by the coefficient of the current row and the candidate row.

$$w_{ik} = | C_{ik} | / \sum_{t=i\pm1,i\pm2} | C_{ik} |. \tag{22}$$

Thus, the estimate for a missing element is computed by,

$$x_{ij} = \sum_{k=i\pm1,i\pm2} x_{kj} w_{ik}. \tag{23}$$

where C_{ik} is the correlation coefficient of row i and k.

3) *Multi-channel Singular Spectrum Analysis (MSSA)*

MSSA is often used to solve missing data problems, e.g., geographic data and meteorological data. It is a data adaptive and nonparametric method based on the embedded lag-covariance matrix. We adopt an iterative procedure proposed in [16] that utilizes the internal periodicity of traffic conditions.

C. *Comparisons*

The four algorithms are compared in terms of estimation error. In Naïve KNN, K is set to 4, in the correlative KNN, K is also set to 4, and in MSSA, the parameter, M, is set to 24 as suggested by [16]. According to the result of Algorithm 2, we set r and λ in Algorithm 1 to 2 and 100, respectively.

In Fig. 11, the performance of the four algorithms in terms of estimation error is shown. Three time granularities are used, i.e., 15 min, 30 min and 60 min. We can see that our algorithm performs the best among all the algorithms under every time granularity. Naïve KNN performs the worst. Correlation-based KNN and MSSA are better than naïve KNN, but worse than our algorithm. The two algorithms, correlation-based KNN and MSSA, produce almost similar performance with respective to estimation error.

We can also find that when the integrity of the traffic condition matrix decreases, our algorithm steadily produces low estimation errors. That is, the performance of our algorithm is relatively insensitive to the integrity of measurement matrices. Even when the integrity is as low as 20%, the estimation error is no more than 20% when the time granularity is 60 minutes. This shows that our algorithm can reliably recover the missing elements when just a few elements are

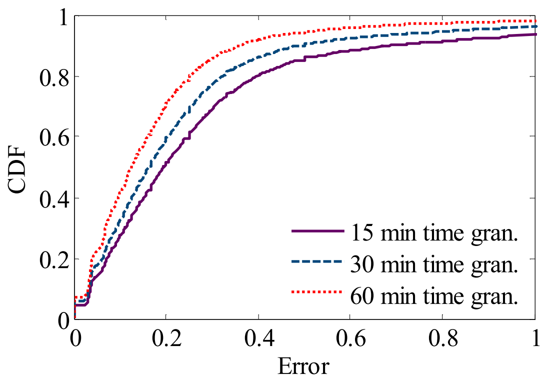


Figure 12. CDFs of relative errors with different time granularities.

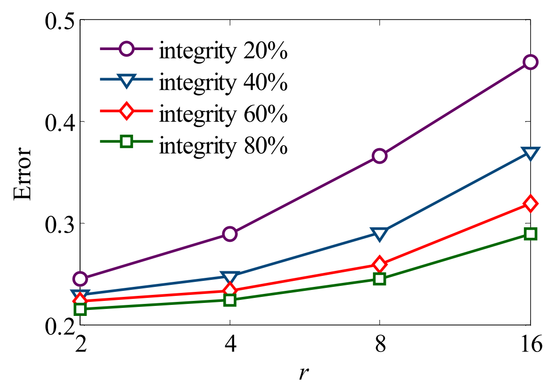


Figure 13. Estimation error against rank bound r ($\boldsymbol{\lambda}$=1, gran.=30min).

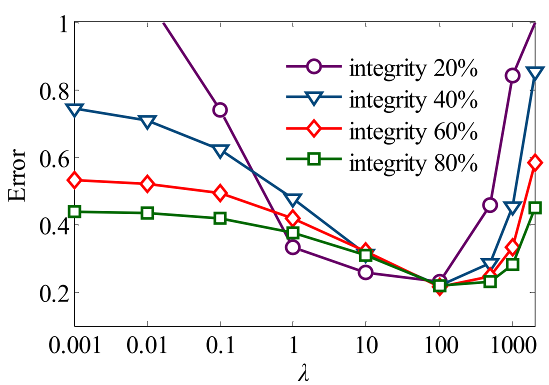


Figure 14. Estimation error against tradeoff coefficient λ ($\boldsymbol{r}$=32, gran.=30min).

available. In contrast, the rest algorithms including naïve KNN, correlation-based KNN and MSSA have worse performance when the integrity becomes poorer. The reason is that our compressive sensing based algorithm can effectively capture the internal structures that exist in the dataset even just a few data points are used, while the rest algorithms fail to achieve this.

From Fig. 11, we can also see that the estimation error becomes higher when the time granularity is smaller for all algorithms. The main reason is that the traffic condition matrix contains more details when the time granularity is smaller. Hence, it is more difficult for algorithms to estimate the original values without losing the details.

We further show the distribution of individual errors. Since absolute errors may differ dramatically, we instead study relative errors. A relative error of an estimated element is defined as $|\hat{x}_{ij} - x_{ij}|/x_{ij}$. The experiments are conducted with integrity of 20%. We can find that 80% of estimated elements have a relative error of smaller than 0.25 when the time granularity is 60 minutes. Even when the time granularity is 15 minutes, the relative error for nearly 80% of estimated elements is less than 0.38.

D. *Impact of Parameters*

As mentioned before, Algorithm 1 has two important parameters, i.e., rank bound and tradeoff coefficient. The parameters impact the algorithm's performance. We have proposed the genetic-based algorithm for finding the optimal parameters. In the following, we conduct experiments to study the impact of these parameters and show that it is important to design the algorithm for finding the optimal parameters.

First, we study the impact of rank bound r. Fig. 13 plots error rate against different rank bounds. In this experiment, the time granularity is 30 minutes and λ is set to one. We find that the estimation error is lowest when the rank bound is two. The main reason is that when the rank of $\hat{X}$ is low, the estimate matrix embodies the major trend of variation of the original matrix. When the rank of $\hat{X}$ grows, the estimate matrix tries to describe more information but is often misled by measurement errors. This increases the estimation error.

We also study the impact of tradeoff coefficient λ. For ease of studying its impact, we set rank bound r to 32. In Fig. 14, estimation errors against different tradeoff coefficients are shown. We find that the estimation error changes significantly when the tradeoff coefficient changes from 0.001 to 2000. The optimal coefficient is around 100 when the rank bound is 32. According to (17), a larger λ puts more weight to rank minimization and a smaller λ more emphasizes measurement fitness. A good tradeoff coefficient should strike a balance between rank minimization and measurement fitness

VI. Related Work

Close-circuit cameras and vehicle loop detectors are two traditional methods for estimating traffic conditions. By installing cameras in road intersections, we can analyze the video screen manually, or by image processing, to estimate traffic conditions at the locations where the cameras are installed. Obviously, it suffers the coverage problem and is limited by the complexity of image processing algorithms. A more common method is deploying inductive circuits under the road surface. When a vehicle passes above, it produces a signal. According to the time interval of two consecutive signals, we can calculate the speed of this vehicle and evaluate the number of vehicles on the road. It suffers the coverage and cost problem as well.

As used in this paper, deploying probe vehicles is a more recent method for traffic sensing. Sensors are placed in vehicles to collect traffic information. Since probe vehicles are driving throughout the city, they are able to collect traffic information over a vast area. This overcomes the coverage problem. Another advantage is that even if some probe vehicles fail for some reasons, it can still work and therefore it is robust. Unlike other wireless mobile sensor networks, probe vehicles do not have strict energy constraints.

In [8], the authors discuss the architecture of probe vehicle systems and develop a simple analytical/statistical model. They figure out that 3% penetrate is needed on highway and over 5% penetrate is required on surface roads. J. Yoon et al [14] focus on how to figure out the street traffic states on a given road segment based on probe vehicle's trace data. They drive a car with a GPS device in a given route in Ann Arbor and collect GPS information every 4 to10 seconds. They classify traffic states according to vehicle's spatial average speed and temporal average speed. Such a method, however, requires sufficient data and each road segment is analyzed independently. They also show that the driving speed of one road segment exhibits some regular patterns. Virtual Trip Lines [10] protect the privacy of individual traces. SEER [16] studies the redundancy of traffic data and recover missing values using Multiple Singular Spectrum Analysis (MSSA).

In summary, although some algorithms have been proposed for solving the problem of missing values in datasets, few of them can effectively exploit the hidden structures in traffic condition datasets as the algorithm proposed by this paper.

VII. Conclusion and Future Work

This paper has presented our new approach based on compressive sensing to large scale traffic sensing in an urban environment. With principal component analysis, we have mined a large amount of trace datasets collected in Shanghai, China, and discover that traffic condition datasets often embody hidden structures or redundancy. According to this finding, we have designed the efficient algorithm based on compressive sensing, which effectively exploits the internal structures of traffic condition matrices and makes accurate estimation. Trace-driven experiments have verified that the algorithm outperforms other algorithms, such as KNN and MSSA. More surprisingly, even when 80% of original data are not available, the algorithm can still achieve an estimation error of as low as 20%. The results suggest that traffic sensing in a large scale metropolis like Shanghai can still be effective even when the number of probe vehicles is not large.

There are three avenues for future study. First, the current work constructs the traffic condition matrix for given locations and given times. However, it is possible to construct different matrices for estimating traffic conditions at different locations or/and times. It is an interesting and important problem to find the best way for constructing adaptive measurement matrices. Second, the current work adopts a centralized method of traffic sensing and future work will explore distributed algorithms for traffic sensing. Finally, the relationship between number of probe vehicles and quality of traffic sensing should also be studied.

Acknowledgment

This research is supported by NSFC (No. 60903190, 61027009, 60970106, 60673166 & 60803124), Shanghai Pu Jiang Talents Program (10PJ1405800), Shanghai Chen Guang Program (10CG11), 973 Program (2005CB321901), MIIT of China (2009ZX03006-001-01 & 2009ZX03006-004), Doctoral Fund of Ministry of Education of China (20100073120021), 863 Program (2009AA012201). In addition, it is partially supported by the Open Fund of the State Key Laboratory of Software Development Environment (Grant No. SKLSDE-2010KF-04), Beijing University of Aeronautics and Astronautics.

References

[1] H. Bar-Gera, "Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel," Transportation Research Part C: Emerging Technologies, vol. 15(6), pp, 380-391, 2007.

[2] R. Baraniuk, "Compressive sensing," IEEE Signal Processing Magazine, vol. 24(4), pp, 118-121, July 2007.

[3] M. Bramberger, J. Brunner, B. Rinner, and H. Schwabach, "Real-time video analysis on an embedded smart camera for traffic surveillance," Proc. IEEE RTAS, 2004.

[4] E. Candes and B. Recht, "Exact matrix completion via convex optimization," Foundations of Computational Mathematics, vol. 9(6), pp, 717-772, 2009.

[5] E. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," IEEE Transactions on Information Theory, vol. 52(12), pp, 5406-5425, 2006.

[6] B. Coifman, "Estimating travel times and vehicle trajectories on freeways using dual loop detectors," Transportation Research: Policy and Practice, vol. 36(4), pp, 351-364, 2002.

[7] D. Donoho, "Compressed sensing," IEEE Transactions on Information Theory, vol. 52(4), pp, 1289-1306, 2006.

[8] M. Ferman, D. Blumenfeld, and X. Dai, "An analytical evaluation of a real-time traffic information system using probe vehicles," Journal of Intelligent Transportation Systems, vol. 9(1), pp, 23-34, 2005.

[9] B. Gao and B. Coifman, "Vehicle identification and GPS error detection from a LIDAR equipped probe vehicle," Proc. ITSC, 2006, pp. 1537-1542.

[10] B. Hoh, M. Gruteser, R. Herring, J. Ban, D. Work, J. Herrera, A. Bayen, M. Annavaram, and Q. Jacobson, "Virtual trip lines for distributed privacy-preserving traffic monitoring," Proc. ACM MobiSys, 2008.

[11] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. Kolaczyk, and N. Taft, "Structural analysis of network traffic flows," ACM SIGMETRICS Performance Evaluation Review, vol. 32(1), pp, 61-72, 2004.

[12] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," SIAM Review, 2007.

[13] B. Recht, W. Xu, and B. Hassibi, "Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization," Proc. IEEE Conference on Decision and Control, 2008, pp. 3065-3070.

[14] J. Yoon, B. Noble, and M. Liu, "Surface street traffic estimation," Proc. ACM MobiSys, 2007, p. 232.

[15] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," ACM SIGCOMM Computer Communication Review, vol. 39(4), pp, 267-278, 2009.

[16] H. Zhu, Y. Zhu, M. Li, and L. Ni, "SEER: Metropolitan-scale traffic perception based on lossy sensory data," Proc. IEEE INFOCOM, 2009.