Compressive Data Retrieval with Tunable Accuracy in Vehicular Sensor Networks

Ruobing Jiang[1], Yanmin Zhu[1,2], Hongjian Wang[1], Min Gao[3], and Lionel M. Ni[1,4]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Shanghai Key Lab of Scalable Computing and Systems
[3] Guangzhou HKUST Fok Ying Tung Graduate School
[4] Hong Kong University of Science and Technology
{likeice,yzhu,hwang}@sjtu.edu.cn, mingao@ust.hk, ni@cse.ust.hk

Abstract. On-demand data retrieval is a crucial routine operation in a vehicular sensor network. However, on-demand data retrieval in a vehicular environment is particularly challenging because of frequent network disruption, large number of data readings and limited transmission opportunities. Real world vehicular datasets usually contain a lot of *data redundancy*. Motivated by this important observation, we propose an approach called *CDR* with compressive sensing for on-demand data retrieval in the highly dynamic vehicular environment. The distinctive feature of CDR is that it supports *tunable accuracy* of data collection. There are two major challenges for the design of *CDR*. *First*, the sparsity level of the vehicular dataset is typically unknown beforehand. *Second*, it is even worse that the sparsity level of the dataset is changing over time. To combat the challenge posed by time-varying data sparsity, *CDR* can terminate from further collection of measurements, based on an adaptive condition on which only localized measurements and computation are needed. Extensive simulations with real datasets and real vehicular GPS traces show that our approach achieves good performance of data retrieval with user-customized accuracy.

Keywords: vehicular sensor networks, data retrieval, compressive sensing, tunable accuracy.

1 Introduction

Thanks to the rapid advance in embedded senors and inter-vehicle radio communications such as Dedicated Short-Range Communications, vehicular sensor networks (VSNs) [1] [2] have recently attracted growing attention from academy, industry and government. Equipped with on-board sensors such as camera, GPS receiver, and 3D accelerometer, mobile vehicles become powerful mobile sensors. As vehicles may move around in a large area and visit different places, they are able to collect useful sensing data that are geo-distributed vastly. Many applications of vehicular sensor networks have been developed, such as road traffic estimation, road surface monitoring, and proactive urban surveillance [3].

K. Ren et al. (Eds.): WASA 2013, LNCS 7992, pp. 426–439, 2013.
© Springer-Verlag Berlin Heidelberg 2013

A vehicle sensor network consists of a lot of self-organizing mobile vehicles. On each vehicle, an array of useful sensors are embedded onboard. As a vehicle is moving around, the sensors can continuously collect sensing data, e.g., air pollution data. When a vehicle is within the communication range of other vehicles, it can share its sensing data with others. A vehicle that wants to learn about the air pollution distribution in the city can get such sensed air pollution data from other vehicles.

On-demand data retrieval is therefore a crucial routine operation in a vehicular sensor network, with which a querying vehicle retrieves all sensing data of the other vehicles. However, on-demand data retrieval in a vehicular environment is difficult because the following characteristics of vehicular ad hoc networks. *First*, there are a large number of vehicles distributed in a vast area. As a result, retrieving all data from the vehicles is very difficult and costly. *Second*, the whole network is typically disconnected and the network topology is highly dynamic. There is no connected path between each pair of vehicles. *Third*, data transmissions between vehicles can only rely on opportunistic inter-vehicle encounters.

There are some existing studies for improving the performance of data retrieval in vehicular sensor networks. A few studies [4] [5] [6] propose to exploit the fixed infrastructure of roadside units or access points to increase inter-vehicle encounter opportunities and then to improve the data retrieval performance. However, the cost of deploying such a large-scale infrastructures is prohibitively high.

Network coding based approaches [7] [8] have also been proposed for increasing data retrieval performance in vehicular environment. The main idea is to encode original sensed data items into encoded blocks by linearly combining these data items with random coefficients. This enables the quick spread of original sensed data and improves the data accessibility. When the number of collected encoded blocks is close to the number of original sensed data items, the Gaussian elimination technique is adopted to decode all the original data items. However, in order to decode all the sensed data items, a large number of encoded blocks should be collected, which would incur a large retrieval delay and high transmission overhead.

Recently, *compressive sensing* [9] has gained growing importance for its unique capability to recover *redundant* or *sparse* data with only a very limited set of measurements. Mathematically, a dataset is sparse if the number of non-zero values is much smaller than the number of all values in the dataset. A dataset may not be sparse in the original domain. However, the dataset is still sparse as long as there exists a domain in which the transformed dataset is sparse.

Motivated by the observation that real world vehicular datasets are usually sparse, we employ compressive sensing for on-demand data retrieval in the highly dynamic vehicular environment. The main idea is to proactively distribute projected data among vehicles after the original data are generated. A retrieving node can then collect a small set of projected data from those encountered vehicles, with which it estimates the original data with high accuracy by applying the technique of compressive sensing.

Unfortunately, two major challenges must be addressed before the compressive sensing based approach is effective. *First*, the sparsity level of the vehicular

dataset is typically unknown beforehand. Second, it is even worse that the sparsity level of the dataset is changing over time. Such time-varying feature of the data sparsity induces great challenges to designing efficient compressive data retrieval approaches.

In this paper, we propose an approach called *CDR* for efficient data retrieval in vehicular sensor networks. *The distinctive feature of CDR is that it supports tunable accuracy of data collection.* To combat the challenges posed by time-varying data sparsity, *CDR* incrementally collects the set of projected data, based on which it recovers the vehicular sensing data. To terminate from further collection of projected data, *CDR* devises an adaptive condition on which only localized measurements and computation are needed. We have performed extensive simulations with real datasets and real vehicular GPS traces. The simulation results show that our approach achieves good performance of data retrieval and can successfully recover vehicular readings with a user-customized accuracy.

The main technical contributions that we have made in the paper are as follows.

- This is the first work, to the best of our knowledge, that deals with the changing sparse level of datasets for compressive sensing based on-demand data retrieval in vehicular sensor networks.
- We have proposed *CDR*, an approach for data retrieval in highly dynamic vehicular environments. With *CDR*, each querying node can retrieve network-wide data retrieval with tunable accuracy. A condition is devised for each querying node to terminate from further collection of projected data, and the condition can be fully evaluated based on localized measurements and computation.
- We have performed extensive simulations based on real vehicular datasets of real vehicular GPS traces, and simulation results show that our approach achieves good performance of data retrieval.

The rest of the paper is organized as follows. The next section reviews related work. Section 3 presents the system model, introduction of basic compressive sensing and problem statement. The basic idea is given in Section 4 and the design details of *CDR* are elaborated in Section 5. The performance evaluation is presented in Section 6. Finally, we conclude the paper in Section 7.

2 Related Work

Previous work on data retrieval in vehicular sensor networks either assume the availability of infrastructures, e.g., road side units [4], or use coding schemes [10] to improve data availability.

CGP [5] takes advantage of road side units (RSUs) to collect datasets from a vehicular environment. The main design consideration is to efficiently use the precious communication chances between RSUs and vehicles. The approach clusters local vehicles on the same road segment and aggregates data in each cluster. Then the aggregated data in each local cluster are relayed by the cluster head to

reach the nearest RSU. The local data aggregation reduces the communication between vehicles and road side units. Thus, the precious upload bandwidth can be efficiently used with less collisions.

DB-VDG [6] considers the vehicular data gathering for a base station under specified time constraints. The base station geocasts its data collection query in its local area. The vehicles in the area around the base station that receive the query collect and transmit their sensed data toward the base station. The query generated by the base station includes a specified time interval to limit the delay of data collection process. The data collection process is active only during the specified time interval.

Such infrastructure based data gathering can not provide the data availability for each individual vehicles. Only RSUs or base stations can get the interested dataset from sensing vehicles. They do not support on-demand data retrieval in vehicular networks.

To support data availability for individual vehicles, new approaches are proposed. Roadcast [11] can provide data availability to each individual vehicle. Roadcast explores the popularity of data to ensure that more popular data are more likely to be shared and have more replicas in the vehicular network. However, Roadcast can only provide the most relevant data to the queries of vehicles. Roadcast can not provide vehicles with the whole data sensed by the whole vehicular sensor network.

Various coding schemes, e.g., network coding and erasure coding, are used to provide accessability to the whole data for each individual vehicle. CodeTorrent [7] and VANETCODE [8] use randomized network coding to combine original data packets generated by vehicles into coded blocks. When the number of collected coded blocks is comparable to the number of all the original data packets, the original data packets can be all recovered. However, the relationship between the number of collected coded blocks and the decodability is not deterministic. In another word, the decodability is not guaranteed even when a large number of coded blocks have been collected.

3 Model and Preliminaries

3.1 System Model

The vehicular sensor network comprises a set of vehicles denoted by $V = \{1, 2, \cdots, N\}$. A vehicle $i \in V$ periodically generates a data reading x_i during the period from the area it travels. Any vehicle $v \in V$ may have the demand to retrieve the set of all data readings $\mathbf{x} = [x_1, x_2, \cdots, x_N]^T$. Vehicles in the sensor network communicate with each other when they are within the communication range of each other.

To increase the data retrieval performance in terms of shorter retrieval delay and higher retrieval accuracy, it is a general approach to introduce two processes. More specifically, the two processes are introduced as follows.

- **Data Replication Process.** Firstly, the *data replication process* starts soon after the dataset is generated. In this process, vehicles proactively distribute

data replication or redundancy. This process lasts for a fixed time length which is called **replication delay** denoted by α. After this process ends, vehicles cannot generate new data replication or redundancy.

– **Data Retrieval Process.** Next, the *data retrieval process* can be started after the data replication processes terminates. When a retrieving node begins to retrieve the dataset, the data retrieval process is started. The retrieving node tries to recover the original dataset by collecting the replicated data from those vehicles that it has encountered, other than each original source vehicle. The time length of the data retrieval process of vehicle i is called **retrieval delay**, denoted by β_i.

The main objectives of efficient data retrieval are as follows. *First*, for any querying vehicle, the retrieval delay should be as short as possible. *Second*, the retrieved dataset should have a high accuracy, meaning that the retrieved dataset should be as close to the original dataset as possible. *Third*, a lower transmission overhead incurred in the whole network is preferred.

3.2 Basics of Compressive Sensing

Compressive sensing enables a potentially large reduction in the sampling for a sparse signal. A signal is sparse if it can be represented using only a few non-zero coefficients in a suitable basis. Then nonlinear optimization can be used to recover such signal with a few samplings.

For a sensing reading vector $\mathbf{x} = [x_1, x_2, \cdots, x_N]^T \in \mathbb{R}^N$, suppose it is sparse in some transform basis $\Phi = \{\phi_i\}_{i=1}^N$ (e.g., wavelet, Fourier) which is usually orthonormal or orthogonal. Then $\mathbf{x}$ can be represented as the product of Φ^{-1} and a sparse *coefficient vector* $\mathbf{d}$,

$$\mathbf{x} = \Phi^{-1}\mathbf{d} = \sum_{i=1}^{N} \phi_i d_i, \tag{1}$$

in which d_i is the coefficient for the basis vector ϕ_i and $\mathbf{d}$ is sparse in terms that the number of non-zero coefficients in $\mathbf{d}$ is small. Moreover, $\mathbf{x}$ is K-sparse if the number of non-zero coefficients is no more than K, i.e., $\|\mathbf{d}\|_0 \leq K$, in which $\|\mathbf{d}\|_0 := |supp(\mathbf{d})|$. When $K \ll N$, we can recover $\mathbf{x}$ with a small number of measurements.

In compressive sensing, a measurement y is a projection (defined as inner product) on the vector $\mathbf{x}$ with a *projection vector* $\psi = [p_1, p_2, \cdots, p_N]^T$, i.e., $y = \psi^T\mathbf{x} = \sum_{i=1}^{N} p_i x_i$. With M measurements, we have the following equation

$$\mathbf{y} = \Psi\mathbf{x} = [\psi_1^T\mathbf{x}, \psi_2^T\mathbf{x}, \cdots, \psi_M^T\mathbf{x}]^T, \tag{2}$$

where $\Psi = [\psi_1, \psi_2, \cdots, \psi_M]^T$ is an $M \times N$ *measurement matrix*. According to [12], the sensing vector $\mathbf{x}$ can be successfully reconstructed when the matrix Ψ satisfies the restricted isometry property (RIP) of order $2K$. Moreover, the

number of measurements M should satisfy the following condition to achieve the RIP given the sparsity level K,

$$M \geq CK \log(\frac{N}{K}), \tag{3}$$

where $C \approx 0.28$.

Given M measurements, the following ℓ_1-norm minimization is solved to construct an estimated $\hat{\mathbf{d}}$ to recover the sparse coefficient vector $\mathbf{d}$,

$$\hat{\mathbf{d}} = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_1, \text{s.t. } \mathbf{y} = A\mathbf{z}, \tag{4}$$

where $A = \Psi\Phi^{-1}$ which is also an $M \times N$ matrix and the ℓ_p norm is defined for $p \in [1, \infty]$ as

$$\|\mathbf{z}\|_p = \begin{cases} \left(\sum_{i=1}^{n} |z_i|^p\right)^{\frac{1}{p}}, p \in [1, \infty)\,; \\ \max\limits_{i=1,2,\ldots,n} |z_i|\,, p = \infty. \end{cases} \tag{5}$$

Finally, the approximation $\hat{\mathbf{x}}$ to $\mathbf{x}$ is constructed with sufficient accuracy when the measurement matrix Ψ holds the RIP,

$$\mathbf{x} = \hat{\mathbf{x}} = \Phi^{-1}\hat{\mathbf{d}}. \tag{6}$$

Note that in practice, we usually use random projection vectors to generate measurements.

4 Basic Idea

We propose to apply compressive sensing to sparse vehicular data retrieval. Our compressive data retrieval (CDR) approach includes two schemes for the two processes introduced in Section 3.1, respectively. In the data replication process, each vehicle shares projections of sensed data with other vehicles once it encounters other vehicles. In such way, the original sensing data generated by each vehicle can be spread over the whole network. In the data retrieval process, a querying vehicle can gather measurements (i.e., data projections) from all the vehicles it encounters. Taking advantage of compressive sensing, the querying vehicle can recover the dataset with only a small number of measurements. The number of measurements is much smaller compared with the size of the dataset.

The key issue of *CDR* is to determine the number of measurements a querying vehicle should take given the user-customized recovery accuracy. In other words, the querying vehicle should determine when to stop the collection of measurements, i.e., the termination condition for the retrieval process. It is challenging because different users may have different demands on the recovery accuracy and retrieval delay. There is a tradeoff between the recovery accuracy and the retrieval delay. According to compressive sensing, more measurements provide higher recovery accuracy. However, in order to improve recovery accuracy by collecting more measurements, a querying vehicle should spend more time. What

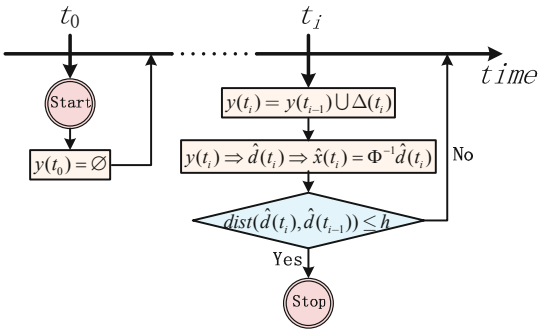

Fig. 1. Illustration of the basic idea. A querying vehicle v raises a data retrieval request at time t_0 and initializes the set of measurements, $\mathbf{y}(t_0)$, as an empty set. At each future time, t_i, when v encounters a set of vehicles, v gathers measurements and decides whether to terminate the data retrieval process.

makes the problem worse is that the sparsity level K of the vehicular dataset is unknown and varies over time. Traditionally, previous approaches usually assume the availability of K and stop the data retrieval process when the number of collected measurements is sufficient (explained in (3)).

Without the knowledge of sparsity level, K, in vehicular environments, we instead evaluate the distance between two sequentially constructed sparse coefficient vectors to decide the termination condition. Because the coefficient vector is the transformed result of the original dataset in another domain, the smaller the distance between two sequentially constructed coefficient vectors, the smaller the distance between two sequentially recovered original datasets. We introduce a threshold of the distance between two sequentially constructed coefficient vectors to adjust the tradeoff between the recovery accuracy and the retrieval delay.

We illustrate the basic idea described above with Fig. 1. Suppose a querying vehicle v queries for the dataset $\mathbf{x}$ at time t_0. v maintains a set of measurements, denoted by $\mathbf{y}$. The set of measurements at t_0 is initialized as an empty set, i.e., $\mathbf{y}(t_0) = \emptyset$.

At each time $t_i > t_0$, the main steps taken by v are as follows. Suppose v encounters a set of vehicles, denoted by $U = \{1, 2, \cdots, u\}, U \subset V$. *First*, it asks each of them to send to it a measurement $y_i, i \in U$. Then v gets a set of measurements $\Delta(t_i)$ from vehicles in U. *Second*, v updates the maintained set of measurements by merging the previous one and $\Delta(t_i)$, i.e., $\mathbf{y}(t_i) = \mathbf{y}(t_{i-1}) \cup \Delta(t_i)$. *Third*, v constructs the sparse vector $\hat{\mathbf{d}}(t_i)$ based on $\mathbf{y}(t_i)$ by solving (4). *Fourth*, i computes the distance between $\hat{\mathbf{d}}(t_i)$ and $\hat{\mathbf{d}}(t_{i-1})$. If the distance is less than a specific threshold h, then v terminates further collections of measurements.

The user-customized threshold h adjusts the tradeoff between the recovery accuracy and the retrieval delay. When a smaller h is specified, i.e., higher recovery accuracy is required by the user v, more measurements are needed, which costs more time.

5 Design of *CDR*

5.1 Overview

In this section, we will introduce the detailed design of our compressive data retrieval (CDR). *CDR* is composed of two main components, namely *data projection scheme* and *data retrieval scheme*.

Data projection scheme is used by each vehicle in the data replication process which specifies how to project sensor readings. We consider the scenario where vehicle i encounters vehicle j at time t and should send a data projection $y_i(t)$ to j. Because j also do the same thing as i, we only consider the actions taken by i. Suppose at time t, the set of projections contained by i is denoted by $\mathbf{y}_i(t)$ and for each $y_s \in \mathbf{y}_i(t)$, the set of sensing data projected in y_s is denoted by $\mathbf{x}_s$.

The main steps are as follows. First, vehicle i selects a subsection $\mathbf{y'}_i(t) = \{y_1, y_2, \cdots, y_r\} \subseteq \mathbf{y}_i(t)$ to make the projection $y_i(t)$. Second, i generates a random coefficient $\psi_i(t)$ for the sensing data x_i. Third, i projects the $y_i(t)$ as follows,

$$y_i(t) = \psi_i(t)x_i + \sum_{s=1}^{r} y_s, y_s \in \mathbf{y'}_i(t). \tag{7}$$

Fourth, i send $y_i(t)$ to j along with the index list of all the sensing data projected in $y_i(t)$ and the corresponding coefficients. Note that the index list of all the sensing data projected in any $y_s \in \mathbf{y'}_i(t)$ and the corresponding coefficients are available because i received them along with the projections y_s.

Initially, when i first encounters a vehicle, it only sends the projection $y_i(t_0) = \psi_i(t_0)x_i$, the index of x_i (i.e., i) and the corresponding coefficient $\psi_i(t_0)$ to the encountered vehicle. We can also find that each $y \in \mathbf{y}_i(t)$ is a projection received by i from other vehicles at time earlier than time t.

There is one key issue for the data projection scheme. To enable the accurate recovery of data vector with high probability for any querying vehicle, the coefficients generated by all the vehicles for their own sensing data should satisfy a condition explained later in Section 5.2.

Data retrieval scheme is designed to decide the termination condition for the retrieval process of a querying vehicle. Suppose the querying vehicle is i and it starts its data retrieval process at time t_0.

Initially, at time t_0, vehicle i initializes its measurement set $\Omega(t_0)$ as empty set and its approximated sensing data vector $\hat{\mathbf{x}}(t_0)$ as zero vector with N elements of zero.

The main steps at time $t > t_0$ are as follows when it encounters one or more vehicles. We only consider one vehicle j that i encounters because each encountered vehicle does the same thing as j. We denote by $\mathbf{y}_z(t)$ the set of projections of vehicle z at time t. *First*, i asks j to send to i a projection of a subset of $\mathbf{y}_j(t)$. *Second*, j makes a projection $y_j(t)$ to send to i using the same projection scheme applied in the data replication process, i.e., the data projection scheme mentioned above. *Third*, i updates the measurement set by adding $y_j(t)$ to $\Omega(t-1)$ to construct the new measurement set $\Omega(t) = \{\omega_s\}_{s=1}^{\pi}$.

Fourth, i approximates the sparse coefficient vector $\hat{\mathbf{d}}(t)$ by solving the following optimization using basis pursuit [13]

$$\begin{aligned} &\hat{d}(t) = \arg\min_{u} \|u\|_1 \\ &\text{s.t. } [\omega_1, \omega_2, \cdots, \omega_\pi]^T = \Psi\Phi^{-1}u, \end{aligned} \tag{8}$$

where $\Psi = [\psi_1^T, \psi_2^T, \cdots, \psi_\pi^T]^T$ is the measurement matrix with ψ_s being the coefficient row vector of sensing data projected in ω_s, and Φ is the Haar wavelet transform basis. Fifth, i evaluates the termination condition by comparing $\hat{\mathbf{d}}(t)$ with $\hat{\mathbf{d}}(t-1)$ and decides whether to collect new measurements or terminate the data retrieval process.

The *key issue* for the data retrieval scheme is to determine the termination condition to balance the tradeoff between the data retrieval delay and the data recovery accuracy.

Next we deal with the key issues in the two schemes.

5.2 Random Gaussian Data Projection

As proved in [14], the $M \times N$ measurement matrix Ψ in (2) holds the RIP [12] with high probability, if the entries ψ_{ij} of Ψ are independent realizations of Gaussian random variables as follows

$$\psi_{ij} \sim \mathcal{N}\left(0, \frac{1}{M}\right), \tag{9}$$

where ψ_{ij} is the entry at ith row and jth column of Ψ.

As a result, in order to make sure the final measurement matrix holds the RIP, the data projection scheme has two rules. First, the coefficients of x_i generated at different time by vehicle i follows the i.i.d. Gaussian random variables as listed in (9). Second, vehicle i selects a subset $\mathbf{y'} = \{y_1, y_2, \cdots, y_r\} \subseteq \mathbf{y}$ to make the projection $y_i(t)$, i.e., $y_i(t) = \psi_i(t)x_i + \sum_{s=1}^{r} y_s$. Suppose the set of sensing data projected in y_s is denoted by $\mathbf{x}_s$. Then the subset $\mathbf{y'}$ should satisfy that any pair of $y_p, y_q \in \mathbf{y'}, \mathbf{x}_p \cap \mathbf{x}_q = \emptyset$. The two rules of our data projection scheme ensures that the coefficients of sensing data in each measurement is the initially generated one, i.e., entries in the measurement matrix of any querying vehicle are i.i.d..

5.3 Termination Condition

The data retrieval scheme solves the issue for each querying vehicle that when to stop the data retrieval process.

Previous work usually assume that the sparsity level K of the sensing data is known. Then a querying vehicle can stop its data retrieval process just when it has collected enough number of measurements given in (3). However, under real environment, we do not have the knowledge of K and the worse thing is that

K varies over time. Instead, our data retrieval scheme observes the sequentially recovered coefficient vector $\hat{\mathbf{d}}$ to decide when to stop with the awareness of the recovery accuracy.

We first explore the relation between the recovery error of $\mathbf{d} = \Phi\mathbf{x}$, denoted by $\varepsilon(\mathbf{d}, \hat{\mathbf{d}})$ and the distance between sequential approximations of $\mathbf{d}$. By $\hat{\mathbf{d}}_M$ we denote the recovered coefficient vector with M measurements.

According to [13], for a sparse vector $\mathbf{d}$ with K non-zero elements, if $\hat{\mathbf{d}}_M + 1 = \hat{\mathbf{d}}_M$ and the entries in the measurement matrix are realizations of a Gaussian random variable, then $\hat{\mathbf{d}}_M = \mathbf{d}$, with probability 1. This proposition holds for any optimization algorithm to the minimization of (4) including basis pursuit which is applied in *CDR*.

As a result, we stop each data retrieval process when the normalized difference ε between two sequentially recovered coefficient vectors is less than a threshold h, i.e., when

$$\varepsilon = \frac{\left\| \hat{d}_M - \hat{d}_{M+T} \right\|_2}{\left\| \hat{d}_M \right\|_2} \leq h. \tag{10}$$

6 Performance Evaluation

6.1 Methodology and Simulation Setup

We conduct extensive trace-driven simulations to evaluate the performance of our compressive data retrieval (*CDR*) with real vehicular sensing datasets. We compare *CDR* with other two data retrieval approaches. The main performance metrics we consider are the data retrieval delay, and data recovery error. The data recovery error is computed as $\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}$, in which $\hat{\mathbf{x}}$ is the approximation of $\mathbf{x}$.

The datasets used in simulations include a real dataset of vehicular speed readings and a synthetic dataset generated with specified sparsity level K. The real dataset were collected from real traces of 2,600 taxis in urban Shanghai, China in January, 2006. Each taxi is equipped with a Global Positioning System (GPS) receiver which periodically reads vehicular positions and speeds.

Two sets of simulations are conducted. The first set of simulations evaluate the performance of three data retrieval approaches under different system parameters including the number of vehicles, data replication delay α and data retrieval delay β. The second set of simulations focus on the effect of the threshold h on the recovery error and retrieval delay of *CDR* approach.

Vehicles move on the road network of urban Shanghai, China and communicate with each other when they are within the communication range of each other. Each time a vehicle encounters another vehicle, only one data projection can be transmitted.

For each simulation setup, 15 runs are conducted and the average results are plotted. The default system parameters are shown in Table. 1.

Table 1. Default System Parameters in Simulations

Parameter	Default Value
Communication Range	150 m
Number of Vehicles	1000
Replication Delay α	4 hours
Retrieval Delay β	2 hours
Threshold h	0.15
Sparsity Level K (synthetic datasets)	10

6.2 Compared Approaches

The two compared data retrieval approaches are two combinations of a simple replication scheme SR and two representative estimation schemes, $k-$NN estimate (KNN) and Gaussian process regression (GP).

- Simple replication (SR). In the data replication process, each vehicle transmits the original sensing data to other vehicles. If a vehicle i has more than one sensing data and encounters another vehicle j, it randomly selects and transmits to j a sensing data that j does not have.
- $k-$NN estimate (KNN). To better recover sensing data, a querying vehicle sets the value of a unavailable sensing data x_i as the average of all the sensing data from k nearest neighboring vehicles of i.
- Gaussian process regression (GP). It interpolate values of a random field at unobserved locations from observations of its value at nearby locations.

The two compared approaches are SR+KNN and SR+GP.

6.3 Performance Comparison

We compare the performance of three approaches under different system parameters for vehicular speed dataset and synthetic dataset.

First, when the number of vehicles varies from 200 to 1,000, the recovery error of all the approach are shown in Fig. 2 and Fig. 5 for speed dataset and synthetic dataset, respectively. *CDR* has the lowest error (lower than 0.4 for speed dataset and lower than 0.2 for synthetic dataset) while the recovery error of SR+KNN and SR+GP stay above 0.9 for both datasets. We can find that when the number of vehicles is extremely small (e.g., 200), the recovery error of *CDR* is much higher than those situations where more vehicles exist. This is because that when less vehicles exist, the encounter chances among vehicles in the network are much lower. Thus, vehicular data are difficult to spread around the whole network which results in the relatively higher recovery error.

Second, when the replication delay α increases from 1 hour to 5 hours, Fig. 3 and Fig. 6 plot the recovery error of all the approaches for different datasets, respectively. For both datasets, *CDR* performs best. In Fig. 3, the recovery error of *CDR* decreases from 0.6 to 0.2. For synthetic dataset, the recovery error of *CDR* decreases below 0.1 when the replication delay is longer than 3 hours.

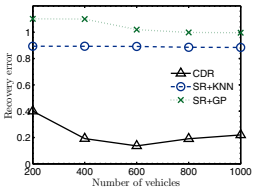

Fig. 2. Recovery error vs. number of vehicles (speed dataset)

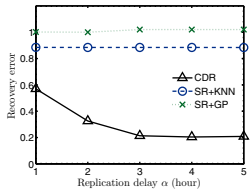

Fig. 3. Recovery error vs. replication delay (speed dataset)

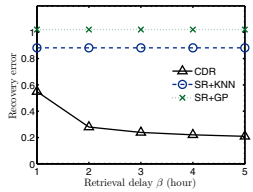

Fig. 4. Recovery error vs. retrieval delay (speed dataset)

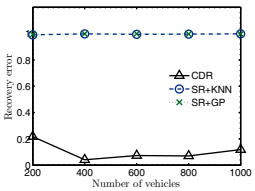

Fig. 5. Recovery error vs. number of vehicles (synthetic dataset)

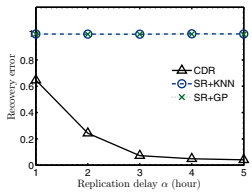

Fig. 6. Recovery error vs. replication delay (synthetic dataset)

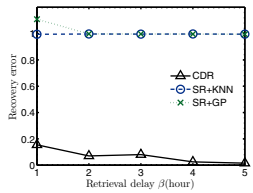

Fig. 7. Recovery error vs. retrieval delay (synthetic dataset)

Third, we present the recovery error of three approaches when the retrieval delay increases from 1 hour to 5 hours in Fig. 4 and Fig. 7. All the compared approaches have lower recovery error as the retrieval delay increases. Moreover, we can find that the recovery error of *CDR* for the synthetic dataset is lower than that for the speed dataset under same retrieval delay. This is because the real speed dataset is not strictly sparse while the synthetic dataset is strictly sparse. There is noise for datasets which is not strictly sparse which results in higher recovery error.

For SR+KNN and SR+GP, the reason why their recovery error keep higher than 0.9 is that the scarce transmission chances can not satisfy the traffic load for all the sensing data to spread over the whole network. By taking advantage of sparsity property, *CDR* can achieve better performance.

6.4 Effect of h

We then explore the effect of threshold h on the performance of *CDR* for a synthetic dataset. Specifically, we evaluate the recovery error, number of collected measurements, and retrieval delay of *CDR*. The sparsity level of the synthetic dataset is set as 10. The threshold h varies from 0.05 to 0.25. Each plotted result is an average value of 15 runs of simulations.

Fig. 8 plots the recovery error of *CDR*. As expected, the recovery error decreases when smaller h is specified. We can also find that the recovery error is

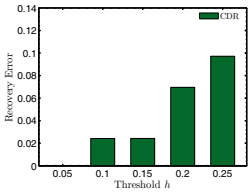

Fig. 8. Recovery error vs. threshold h (synthetic dataset)

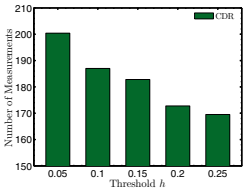

Fig. 9. Number of measurements vs. threshold h (synthetic dataset)

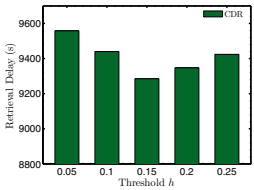

Fig. 10. Retrieval delay vs. threshold h (synthetic dataset)

smaller than h. As a result, user-customized recovery accuracy can be achieved by adjusting the parameter h.

Fig. 9 presents the number of collected measurements when different threshold, h, is specified. As expected, the number of measurements decreases as the bigger h is set. It is reasonable because bigger h means lower recovery accuracy is required which needs less measurements. We can also find that the relation between the number of measurements and the value of h is approximately linear.

Fig. 10 shows the data retrieval delay under different threshold. From the figure, we can not find clear relation between the retrieval delay and h. This is mainly because that the retrieval delay is not only determined by specified recovery error, but also related to the encounter rate between the querying vehicle and other vehicles. When the recovery error is specified or a specified h is given, the number of measurements is also determined. However, the delay to collect a given number of measurements varies for different vehicles. If the vehicle is located in a dense area with high encounter rate with other vehicles, then the delay will be much small. On the other hand, when the vehicle is located in a sparse area with much less vehicles, then the delay to collect a fixed number of measurements will be much longer.

7 Conclusion

In this paper we have focused on the crucial problem of on-demand data retrieval in a highly dynamic but challenged vehicular environment. Inspired by the observation that real vehicular datasets are usually sparse, we have proposed *CDR* for on-demand data retrieval with tunable accuracy. With compressive sensing, *CDR* realizes tunable accuracy by devising a condition on which a retrieving node can test based on local measurements and stops further collection of measurements. This effectively combats the challenge posed by time-varying sparsity of the vehicular sensing datasets. Based on the real vehicular sensing datasets and real vehicular GPS traces collected from around 2,600 taxis in Shanghai, China, our simulation results demonstrate that *CDR* achieves good performance of data recovery accuracy.

Acknowledgements. This research is supported by NSFC (No. 61170238, 60903190, 61027009, 60933011, 61202375, 61170237), Shanghai Pu Jiang Talents Program (10PJ1405800), Shanghai Chen Guang Program (10CG11), MIIT of China (2009ZX03006-001-01), Doctoral Fund of Ministry of Education of China (20100073120021), National 863 Program (2009AA012201 and 2011AA010500), HP IRP (CW267311), SJTU SMC Project (201120), STCSM (08dz1501600, 12ZR1414900), Singapore NRF (CREATE E2S2), and Program for Changjiang Scholars and Innovative Research Team in Universities of China (IRT1158, PCSIRT).

References

1. Lee, U., Gerla, M.: A survey of urban vehicular sensing platforms. Computer Networks 54(4), 527–544 (2010)
2. Zhang, C., Lu, R., Lin, X., Ho, P.H., Shen, X.: An efficient identity-based batch verification scheme for vehicular sensor networks. In: Proc. IEEE INFOCOM, pp. 246–250. IEEE (2008)
3. Lee, U., Zhou, B., Gerla, M., Magistretti, E., Bellavista, P., Corradi, A.: Mobeyes: smart mobs for urban monitoring with a vehicular sensor network. IEEE Wireless Communications 13(5), 52–57 (2006)
4. Yang, L., Xu, J., Wu, G., Guo, J.: Road probing: Rsu assisted data collection in vehicular networks. In: 5th International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2009, pp. 1–4. IEEE (2009)
5. Salhi, I., Cherif, M.O., Senouci, S.M.: A new architecture for data collection in vehicular networks. In: IEEE International Conference on Communications, ICC 2009, pp. 1–6. IEEE (2009)
6. Palazzi, C.E., Pezzoni, F., Ruiz, P.M.: Delay-bounded data gathering in urban vehicular sensor networks. Pervasive and Mobile Computing 8(2), 180–193 (2012)
7. Lee, U., Park, J.S., Yeh, J., Pau, G., Gerla, M.: Code torrent: content distribution using network coding in vanet. In: Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking, pp. 1–5. ACM (2006)
8. Ahmed, S., Kanhere, S.S.: Vanetcode: network coding to enhance cooperative downloading in vehicular ad-hoc networks. In: Proceedings of the 2006 International Conference on Wireless Communications and Mobile Computing, pp. 527–532. ACM (2006)
9. Baraniuk, R.G.: Compressive sensing (lecture notes). IEEE Signal Processing Magazine 24(4), 118–121 (2007)
10. Fujimura, A., Oh, S.Y., Gerla, M.: Network coding vs. erasure coding: Reliable multicast in ad hoc networks. In: IEEE Military Communications Conference, MILCOM 2008, pp. 1–7. IEEE (2008)
11. Zhang, Y., Zhao, J., Cao, G.: Roadcast: a popularity aware content sharing scheme in vanets. ACM SIGMOBILE Mobile Computing and Communications Review 13(4), 1–14 (2010)
12. Davenport, M.A., Duarte, M.F., Eldar, Y.C., Kutyniok, G.: Introduction to compressed sensing. Preprint 93 (2011)
13. Malioutov, D., Sanghavi, S., Willsky, A.: Compressed sensing with sequential observations. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2008, pp. 3357–3360. IEEE (2008)
14. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M.: A simple proof of the restricted isometry property for random matrices. Constructive Approximation 28(3), 253–263 (2008)