Density-Aware Compressive CrowdSensing

Xiaohong Hao*
Tsinghua University
haoxiaohong.ivy@gmail.com

Liwen Xu†
Tsinghua University
kyoi.cn@gmail.com

Nicholas D. Lane
Nokia Bell Labs
University College London
niclane@acm.org

Xin Liu
U.C. Davis
liu@cs.ucdavis.edu

Thomas Moscibroda
Microsoft Research
moscitho@microsoft.com

ABSTRACT

Crowdsensing systems collect large-scale sensor data from mobile devices to provide a wide-area view of phenomena including traffic, noise and air pollution. Because such data often exhibits sparse structure, it is natural to apply compressive sensing (CS) for data sampling and recovery. However in practice, crowd participants are often distributed highly unevenly across the sensing area, and thus the numbers of observations collected over different areas may vary wildly – an issue we call *density disparity*. Density disparity leads to inaccuracy in low density areas, and potentially undermines the recovery performance if conventional compressive sensing is applied directly, which equally treats data from areas of different density.

To address this challenge, we propose a probabilistic accuracy estimator, based on which we devise two recovery algorithms: Threshold Recovery (TR) and Weighted Recovery (WR). As general-purpose recovery algorithms, TR and WR improve the performance of CS in the scenarios with density disparity, and also provide better guarantees in terms of ℓ_2-norm accuracy compared with conventional CS recovery algorithms. We also conduct extensive experiments based on synthetic and real-life datasets. Our results show that TR/WR typically reduce ℓ_2-norm error by more than 60% compared to state-of-the-art baselines.

CCS CONCEPTS

•Information systems →Location based services; Mobile information processing systems; *Data cleaning;* **•Human-centered computing →Ubiquitous computing;**

KEYWORDS

Crowdsensing; Compressive Sensing; Density Disparity

*The work was done when the author was in Tsinghua University. The author is currently affiliated with Google.

†The work was done when the author was in Tsinghua University. The author is currently affiliated with Google.

IPSN 2017, Pittsburgh, PA USA

DOI: http://dx.doi.org/10.1145/3055031.3055081

ACM Reference format:
Xiaohong Hao, Liwen Xu, Nicholas D. Lane, Xin Liu, and Thomas Moscibroda. 2017. Density-Aware Compressive CrowdSensing. In *Proceedings of The 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, Pittsburgh, PA USA, April 2017 (IPSN 2017),* 11 pages.
DOI: http://dx.doi.org/10.1145/3055031.3055081

1 INTRODUCTION

Mobile crowdsensing systems have been attracting increasing attention both in academia and industry. Large-scale distributed data-collection is now easier than ever because of wide adoption of mobile devices. These devices are often sensor-equipped smartphones [13] but can also be more specialized, for example wearables like Google Glass[1] and Narrative Clip[2]. Aggregating this data enables a number of crowd-based systems, for example for group monitoring [24], building noise maps [21], shared video experiences [9] and wireless network signal strength maps[3].

Many popular end-applications in essence require spatial and temporal coverage, such as road conditions [23], mobility models [1], noise and WiFi maps, while limiting the number of samples. Many factors such as competition for limited sensor resources on a single phone or overall transmission burden on network facilities supply incentives of doing more with fewer measurements [8]. Considerable interest exists in determining methods that operate that lower, or eliminate the need for user-based data collection [27] – but this may not be possible, depending on the target scenario.

We consider the specific problem that aims to reconstruct *average* profile from data collected by crowdsensing systems. It is one of the most commonly used data collection model, where the ground truth is the average of the data collected from all the participants in the same area. For example, to construct a temperature heatmap for a city, it is natural to segment the city into many geographic areas and obtain an average temperature estimation for each area. The underlying data of interest (temperature in this example) usually inheres temporal-spatial correlation, and thus can be sparsely represented in some transform domains (e.g. Fourier, DCT). Considering these properties, Compressive Sensing (CS) is a promising technique for the purpose of sampling and reconstructing data at a lower cost.

To apply CS in a crowdsensing system, there are two principal phases: 1) participant data collection, and 2) data processing and

[1]Google Glass: http://www.google.com/glass/start/
[2]Narrative Clip: htttp://getnarrative.com
[3]OpenSignal: htttp://opensignal.com

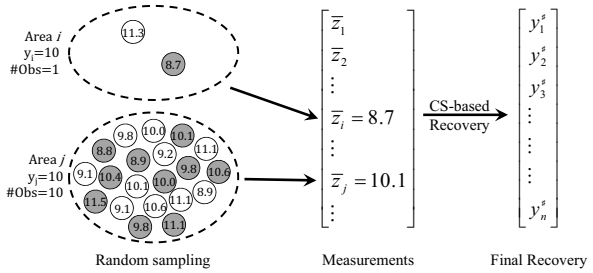


Figure 1: Density issue in crowdsensing systems

signal reconstruction. In the first phase of data collection, standard CS schemes (c.f. [33]) randomly sample a subset of the crowd participants to retrieve their data. Since the population is typically not uniformly distributed spatially or temporally, the numbers of observations for different areas (or time periods) may vary wildly, and further result in highly non-uniform measurement accuracy – we call this issue *density disparity*. Intuitively fewer samples result in worse accuracy and vice verse. Such density disparity and its resulting accuracy disparity are not considered in the conventional CS, which is the focus of this work.

As demonstrated in Fig. 1, two areas have vastly different population (2 vs. 20) and have the same ground truth value 10 of all participants' mean. By randomly sampling the same proportion of the population (50% in this example as colored gray) and taking average of the samples, we can see a massive gap between the measurement values $\bar{z}_i = 8.7$ and $\bar{z}_j = 10.1$. Clearly, these two values have different accuracy levels while the conventional CS treats them equally. Note that *random sampling* is the standard choice of crowdsensing systems because of its simplicity in most practical scenarios. Therefore, we do not alter this widely used sampling method. Instead, we focus on the latter phase – data processing and reconstruction using sample values collected through random sampling.

To address the density disparity and the corresponding accuracy disparity issue, we go one step further than blindly applying conventional CS into crowdsensing systems by augmenting recovery algorithms with the awareness of measurement density. Generally speaking, CS consists of two steps: sensing and recovering. In this work, random sampling is still applied in the sensing step due to its convenience in the implementation. In the recovering step, new algorithms – Threshold Recovery(TR) and Weighted Recovery (WR) are developed to tackle the density disparity issue, aiming for better performance guarantee, which is the ultimate goal of CS. Summing up, we not only apply CS into the crowdsensing system, but also extend CS itself in general.

In this paper, we propose the system of Density-Aware Compressive CrowdSensing(DACCS), specifically designed for these mobile crowdsensing applications. To the best of our knowledge, we are the first to explore the density disparity and the resulting accuracy disparity issue in compressive crowdsensing systems. Specifically, our contributions are:

- Density disparity undermines the performance of conventional CS, which assumes the measurement accuracy are the same. To address the challenge, we propose two density-aware algorithms – TR and WR as alternatives for conventional CS recovery algorithm, e.g. Basis Pursuit. TR is an algorithm motivated by the advantage of throwing away the most noisy measurements. WR generalizes TR by assigning weights inversely proportional to the accuracy estimator. The physical meaning of WR is likelihood maximization. Theoretically, both algorithms provide ℓ_2-norm error guarantees, outperforming conventional CS recovery.
- We study the density disparity issue by bridging non-uniform density with its corresponding measurement accuracy. The notion of measurement accuracy is formulated as the variance of the observation mean, and we propose a high-probability upper bound as the accuracy estimator. The estimator provides necessary information for the recovery algorithms TR and WR.
- Equipped with the density-aware recovery algorithms, we propose the framework of DACCS which applies CS into crowdsensing system to reduce the cost for data collection, while achieving a better recovery performance than conventional compressive crowdsensing systems. The system is consisted of two parts, corresponding to the two steps in CS. In the sampling/sensing step, a small fraction of data are randomly sampled from all the participants. Then our proposed CS recovery algorithms TR and WR are applied in the system for data recovery.
- The performances of TR/WR are evaluated both in an ideal scenario with synthetic data and in a practical scenario with a real-life traffic monitoring dataset. In both experiments, our algorithms outperform conventional CS baselines significantly by more than 60% in terms of ℓ_2-norm error.

The rest of the paper is organized as follows: Section 2 introduces the fundamentals of CS with notations. In Section 3, we present the motivations and challenges in using CS in crowdsensing systems with density disparity, and propose a probabilistic estimator that upper bounds measurement accuracy. Section 4 presents TR and WR to improve recovery accuracy and shows their performance guarantees. Section 6 presents evaluative results, via both numerical experiments and a simulation of a practical scenario. Finally Section 7 and 8 discuss future research directions, related work and conclude the paper.

2 BACKGROUND

In this section we revisit concepts and mechanisms of crowdsensing, and also preliminaries of CS.

2.1 Crowds meet Compressive Sensing

Mobile crowdsensing is a process where people contribute data from personal sensor-equipped mobile devices to some often-centralized cause. Typically, a central controller divides some task into subtasks and assigns these to the participants of the crowdsensing system. Participants may provide data gathered from their devices voluntarily or at the request of controller. The data is then processed by the controller for a particular purpose, usually a benefit to the public

(e.g. finding a WiFi hotspot) or for social welfare (e.g. pollution monitoring).

One challenge in crowdsensing systems is the burden on participants. For example, people may have concerns about their phone battery, network fee or personal privacy. Given these problems, related research often focuses on how to reduce individual cost in these systems, for example, by minimizing effects on battery life [13], or by designing incentive mechanisms [31]. Perhaps the most straightforward solution is to reduce the number of users who must contribute data.

CS is an excellent way of retrieving high-quality data with far fewer data points than alternatives. Since its inception a decade ago, CS has spread to a diverse set of applications, most prominently in medical imaging [16] and networking [32]. In particular, CS has been increasingly employed in sensor network applications [15, 28], as these targets often have an inherent sparse nature. Given the sparsity of these problems, CS is also a promising tool for crowdsensing systems specifically [33].

2.2 Compressive Sensing Preliminaries

CS is an emerging field on the revelation that a relatively small number of non-adaptive linear combinations of a compressible or sparse signal contains enough information for recovery. Suppose $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ is a *k-sparse* vector, i.e. $\mathbf{x}$ contains no more than k nonzero elements where $k \ll n$. CS recovers $\mathbf{x}$ from the linear projections $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is an $m \times n$ measurement matrix, $m \ll n$ and satisfies the critical "Restricted Isometry Property" (RIP) requirement stated below.

Definition 2.1 (RIP requirement[2]). A matrix $\mathbf{A}$ satisfies the restricted isometry property (RIP) of order k if there exists $\delta_k \in (0, 1)$ such that

$$(1 - \delta_k)\|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1 + \delta_k)\|\mathbf{x}\|_2^2, \tag{1}$$

for all k-sparse vector $\mathbf{x}$. δ_k is called the k-restricted isometry constant for matrix $\mathbf{A}$.

In this paper, the measurement matrix $\mathbf{A}$ is constructed from an orthogonal base matrix $\mathbf{B}$ (e.g. Fourier base, DCT base). Define I_i as the indicator of choosing the i^{th} row from $\mathbf{B}$. Let $\Omega = \{i : I_i = 1\}$ be the set of indices sampled via a Bernoulli process, with success probability $\Pr[I_i = 1] = \frac{m}{n}$. Let $\mathbf{B}_\Omega$ denote the submatrix of $\mathbf{B}$ by taking the rows indexed by Ω. When $\mathbb{E}|\Omega| = m = O(k \log^4 n)$, $\mathbf{B}_\Omega$ satisfies RIP with probability $1 - 5e^{-c}$[7].

If Ω is drawn by a non-uniform Bernoulli process, i.e. $\Pr[I_i = 1] = \frac{m_i}{n}$, $\sum_{i=1}^{n} m_i = mn$, the following lemma holds by applying a similar technique as in [25]:

Lemma 2.2 (RIP for non-uniform sampling). *Let $\mathbf{B}_\Omega$ be a measurement matrix constructed from orthogonal base matrix $\mathbf{B}$. With probability $1 - 5e^{-c}\kappa$, $\mathbf{B}_\Omega$ satisfies RIP with $\delta_{3k} + 3\delta_{4k} < 2$, if*

$$\mathbb{E}|\Omega| = m = O(k \log^4 n)$$

where $\kappa = \Pi_{i=1}^{n} \kappa_i$ with

$$\kappa_i = \begin{cases} \frac{m_i}{m} & m_i \geq m \\ \frac{n - m_i}{n - m} & otherwise \end{cases}$$

In Lemma 2.2, δ_{3k} and δ_{4k} are $3k$ and $4k$-restricted isometry constants for matrix $\mathbf{B}_\Omega$, referring to Definition 2.1.

When matrix $\mathbf{A}$ satisfies RIP, a k-sparse vector $\mathbf{x}$ can be *exactly* recovered from $\mathbf{y}$ by solving the ℓ_1-norm minimization problem (P_0).

$$(P_0) \quad \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t. } \mathbf{A}\mathbf{x} = \mathbf{y}$$

It can be further shown that even if the linear measurements $\mathbf{y}$ are perturbed by some bounded additive random noise, $\mathbf{x}$ can still be recovered with bounded error by solving (P_1), a relaxed version of (P_0).

$$(P_1) \quad \min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{s.t. } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon$$

This is formalized as the following lemma,

Lemma 2.3 ([3]). *Suppose measurement matrix $\mathbf{A}$ satisfies RIP with $\delta_{3k} + 3\delta_{4k} < 2$. Then for any signal $\mathbf{x^0}$ that is k-sparse, and its noisy measurement $\mathbf{y} = \mathbf{A}\mathbf{x^0} + \mathbf{e}$, where $\|\mathbf{e}\|_2 \leq \epsilon$, the solution $\mathbf{x}^\sharp$ to (P_1) obeys*

$$\|\mathbf{x}^\sharp - \mathbf{x^0}\|_2 \leq C_k \cdot \epsilon$$

where C_k is a constant dependent only on δ_{4k}. For reasonable values of δ_{4k}, C_k is well-behaved; e.g. $C_k \approx 8.82$ for $\delta_{4k} = 0.2$ and $C_k = 10.47$ for $\delta_{4k} = 0.25$.

3 MOTIVATION AND ERROR APPROXIMATION

As discussed in the introduction, the density disparity within large-scale crowdsensing systems inevitably results in non-uniformity of measurement accuracy – the higher the density, the higher the accuracy. In this section, we first introduce general notation we will use throughout the paper and then present a simple example on how we rescue CS-based recoveries from non-uniformity of measurement accuracy. The example sheds some light on "the power of keeping less to make more", which motivates our formulation of an estimator for measurement accuracy to rise to the most challenging task while lacking prior knowledge of the ground truth. Furthermore, the estimator is proven to upper bound the variance of observation mean with high probability, and it is used as a building block of our two main algorithms in the next section.

3.1 Notation

Vector $\mathbf{y}$ is a signal of size n that the crowdsensing system aims to obtain. This $\mathbf{y}$ can be represented as the product of a orthogonal base matrix $\mathbf{B}$ and a vector $\mathbf{x}$, i.e. $\mathbf{y} = \mathbf{B}\mathbf{x}$. Here $\mathbf{x} \in \mathbb{R}^n$ is a k-sparse vector, where $k \ll n$. For each entry y_i of $\mathbf{y}$, we make d_i independent observations $\mathbf{z_i} = \{z_{i,1}, z_{i,2}, \ldots, z_{i,d_i}\}$, where $z_{i,j}, \forall j \in \{1, 2, \ldots, d_i\}$ are i.i.d. and obey a distribution with mean y_i and variance σ_i^2. Given the observations $\mathbf{Z} = \{\mathbf{z_1}, \mathbf{z_2}, \ldots, \mathbf{z_n}\}$, our goal is to recover $\mathbf{y}$. Following the terminology in conventional CS, measurement value $\bar{z_i}$ of i^{th} entry is an estimation for y_i from multiple observations $\mathbf{z_i}$, in particular, $\bar{z_i} = \frac{1}{d_i} \sum_{j=1}^{d_i} z_{i,j}$.

3.2 More with Less: The Power of Throwing Away

In Section 2.2 we showed that conventional CS provides a theoretical guarantee for signals with bounded noise. As shown in Lemma 2.3, as long as the RIP requirement is satisfied, the ℓ_1-norm

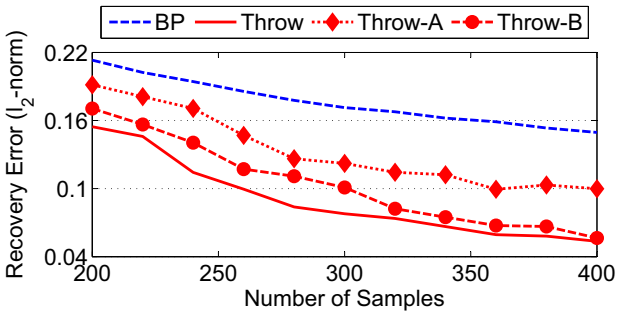


Figure 2: The power of throwing away

minimization recovery's accuracy (in terms of ℓ_2-norm error) heavily depends on the noise level of the provided samples (via the noise bound ϵ). This inspires a simple heuristic to increase recovery accuracy: since we can discard many samples while retaining the RIP requirement (and retaining guarantees on recovery accuracy), we could actually improve recovery quality by *throwing away* the noisiest samples. In terms of Lemma 2.3, we reduce the noise bound ϵ without drastically affecting C_k.

Though throwing away data to better recover data may seem counter-intuitive, in Fig. 2 we demonstrate the effect of discarding noisy samples on recovery error on air pollution data. Standard CS method (Basis Pursuit)[6], keeping all samples, produces the worst error curve, while discarding all samples with error higher than a fixed threshold is the best (approximately 15% of the samples with largest errors are discarded). The recovery error is reduced by approximately a half at 250 samples in this scenario. The other two curves show the ill effects of keeping noisy samples. The curve Throw-A is generated the same as the standard thresholding, except each sample with large error has a 10% chance of not being discarded, while for Throw-B, samples of small error are wrongly thrown away with a 10% chance each. It can be observed from the curves that it is beneficial to *aggressively* discard samples with high error, even if some potentially accurate samples are lost.

This phenomenon is actually predictable and can be formalized per Lemma 2.3. Since samples with large error are removed, the average ℓ_2-norm error of remaining samples is smaller, thus leading to a tighter error bound.

Now recall the concept of the density disparity issue in a crowdsensing system, where sample values are determined by multiple observations. In our setting it is also natural to assume that the noise levels vary spatially and temporally. For example, in the traffic monitoring system [33], the sample value – average speed – is computed by the speed reports from multiple taxis in a road segment. The sample noise, namely the sample variance, varies by road condition and road type. During rush hours, when there is a traffic jam, the speed variance is small since most vehicles are at a very low speed. Conversely, during off-peak hours, individual speed varies largely due to vehicle condition, driver's habit etc.

The cause of this noise is twofold: 1) the underlying distribution (or population distribution) the observations are drawn from, and 2) the number of observations. Understanding these sources of noise justifies the observed *power of throwing away* in CS-based crowdsensing systems.

Although throw-away boosts the recovery accuracy significantly, it depends on the knowledge of *actual* error between each sample value and its true value, which is unavailable beforehand in practical applications. We circumvent this difficulty by using an approximation that neatly estimates the measurement error level. In the throw-away process, it is essential to estimate the sample value and error level for each entry y_i from the observations $\mathbf{z_i} = \{z_{i,1}, z_{i,2}, \dots z_{i,d_i}\}$. We use the observed mean and variance to approximate the sample value and error level. However, when the number of observations is small, this approximation could be wildly inaccurate. To solve this, we show an upper bound for the approximation in terms of ℓ_2-norm error and apply it to the throw-away process.

3.3 Accuracy Estimator

Without ground truth data it is of course impossible to know which sample means $\bar{z_i} = \frac{1}{d_i}\sum_j z_{i,j}$ (calculated from our observations) stray farthest from the target underlying mean y_i. Instead, we will derive a high-probability upper bound on the variance of $\bar{z_i}$, and use this to formulate our algorithms in the next section.

By the Law of Large Numbers, as $d_i \to \infty$, $\bar{z_i} \to y_i$ with probability 1. From the Central Limit Theorem, for large d_i we can approximate the distribution of $\bar{z_i}$ as a normal distribution with mean y_i and variance $\frac{\sigma_i^2}{d_i}$, which indicates

$$\mathrm{Var}(\bar{z_i}) = \frac{\sigma_i^2}{d_i} \approx \frac{s_i^2}{d_i}, \tag{2}$$

where $s_i^2 = \frac{1}{d_i - 1}\sum_j (z_{i,j} - \bar{z_i})^2$ denotes the sample variance of $\mathbf{z_i}$, which is an unbiased and consistent estimator of σ_i^2.

However, when d_i is small, the estimators $\bar{z_i}$ and s_i^2 are more likely to be far from their real values. Thus we take $\mathrm{Var}(s_i^2)$ into consideration in estimating the standard deviation of $\bar{z_i}$. Since s_i^2 is an unbiased estimator of σ_i^2, we have $\mathbb{E}(s_i^2) = \sigma_i^2$, and by Chebyshev's inequality, for all $\eta > 0$,

$$\Pr\left[|s_i^2 - \sigma_i^2| \le \sqrt{\frac{\mathrm{Var}(s_i^2)}{\eta}}\right] \ge 1 - \eta. \tag{3}$$

$$\Rightarrow \quad \Pr\left[\sigma_i^2 \le s_i^2 + \sqrt{\frac{\mathrm{Var}(s_i^2)}{\eta}}\right] \ge 1 - \eta \tag{4}$$

$$\Rightarrow \quad \Pr\left[\frac{\sigma_i^2}{d_i} \le \frac{s_i^2}{d_i} + \frac{1}{d_i}\sqrt{\frac{\mathrm{Var}(s_i^2)}{\eta}}\right] \ge 1 - \eta \tag{5}$$

$$\Rightarrow \quad \Pr\left[\mathrm{Var}(\bar{z_i}) \le \frac{s_i^2}{d_i} + \frac{1}{d_i}\sqrt{\frac{\mathrm{Var}(s_i^2)}{\eta}}\right] \ge 1 - \eta \tag{6}$$

Special Case: when the observations $\mathbf{z}_i$ are drawn from normal distributions centered at y_i, then $\mathrm{Var}(s_i)^2$ in Eq. (6) can be computed as

$$\mathrm{Var}(s_i^2) = \left[\frac{d_i - 1}{\chi^2_{1-\eta}(d_i - 1)} - \frac{1}{d_i}\right] \cdot s_i^2\sqrt{d_i\eta}, \tag{7}$$

where $\chi^2_{1-\eta}(d_i - 1)$ stands for the chi-squared value with probability $1 - \eta$ and degree of freedom $d_i - 1$.

General Case: However, if the underlying distribution of $z_{i,j}$ is unknown, computing $\mathrm{Var}(s_i^2)$ remains a challenge. Supposing the

σ_i and κ_i are the standard deviation and kurtosis of the underlying distribution of $z_{i,j}$, we have [11]

$$\mathrm{Var}(s_i^2) \approx \frac{1}{d_i}\sigma_i^4(\kappa_i - 1). \tag{8}$$

The exact value of $\mathrm{Var}(s_i^2)$ cannot be obtained without knowing the underlying distribution, specially the values of σ_i and κ_i. And learning their exact values from historical data are too expensive and time-consuming. In practice, we assume there is an upper bound for σ_i and κ_i for $i = 1 \ldots n$, i.e. $\sigma_i \le \sigma_{\max}$ and $\kappa_i \le \kappa_{\max}$. We could choose a global upper bound

$$\gamma = \sqrt{\sigma_{\max}^4(\kappa_{\max} - 1)} \tag{9}$$

instead of distinct σ_i and κ_i for each entry. Experimental results show that this simplification does not significantly hurt performance. And the value of γ is stable if the underlying distribution does not change dramatically, so in practice, a value for γ could be learned from historical data.

Recall that keeping inaccurate measurements is more harmful than discarding the accurate ones. We use an $(1-\eta)$-upper bound of $\mathrm{Var}(\bar{z}_i)$ as the accuracy estimator. Combining Eq. 6, 8 and 9, the accuracy estimator is defined as

$$\theta_i^2 := \frac{s_i^2}{d_i} + \frac{\gamma}{d_i^{1.5}\sqrt{\eta}} \tag{10}$$

Here θ_i^2 consists of two parts: the first term $\frac{s_i^2}{d_i}$ is the sample variance, and the second term $\frac{\gamma}{d_i^{1.5}\sqrt{\eta}}$ *penalizes* small d_i's. When d_i is small, $\frac{s_i^2}{d_i}$ does not approximate $\frac{\sigma_i^2}{d_i}$ well, however $\pm\frac{\gamma}{d_i^{1.5}\sqrt{\eta}}$ indicates a large η-confidence interval of s_i^2. So the squared error of $\bar{z}_i$ still lies within $[0, \theta_i^2]$ with high probability. Furthermore, θ_i^2 is also a consistent estimation of σ_i^2 because the penalty term vanishes as d_i grows, which means that θ_i^2 still approximates $\frac{\sigma_i^2}{d_i}$ well when d_i is large.

4 ALGORITHM DESIGN

Based on the aforementioned estimator for measurement accuracy, we present two algorithms in this section. The first algorithm – Threshold Recovery(TR) is a straightforward implementation of estimator-based hard thresholding. Then we generalize Threshold Recovery to Weighted Recovery(WR) by viewing the recovery procedure as statistical inference of hidden variables, which we solve by maximum likelihood estimation. We also show theoretical performance guarantees for both TR and WR by giving upper bounds of ℓ_2-norm error.

4.1 Threshold Recovery

Threshold Recovery (TR), as described in Algorithm 1, directly employs θ_i from Eq. (10) as the surrogate of each measurement's accuracy. Larger θ_i indicates higher probability that the $\bar{z}_i$ is far from its true value y_i, which means those measurements should be considered as harmful to the final recovery. In the threshold recovery scheme, we simply set a threshold θ beforehand and discard all the samples with $\theta_i \ge \theta$. By doing so, with high probability the error of the remaining samples is upper bounded by a smaller value $\epsilon_t \le \epsilon$, where ϵ is a $(1-\rho)$-error bound in Lemma 2.3.

Algorithm 1 Threshold Recovery

Input: Observations $\{z_{i,j}\}_{1\le i\le n, 1\le j\le d_i}$, base matrix $\mathbf{B}$, and a threshold $\theta > 0$.
Output: $\mathbf{y}^\sharp$, the estimation of original $\mathbf{y}$.

Step 1: Calculate $\forall i : 1 \le i \le n$,

$$\bar{z}_i = \tfrac{1}{d_i}\textstyle\sum_j z_{i,j},$$
$$s_i^2 = \tfrac{1}{d_i - 1}\textstyle\sum_j (z_{i,j} - \bar{z}_i)^2,$$
$$\theta_i^2 = \tfrac{s_i^2}{d_i} + \tfrac{\gamma}{d_i^{1.5}\sqrt{\eta}}$$

Step 2: Let $\Omega = \{i : \theta_i \le \theta\}$, $\mathbf{B}_\Omega$ the submatrix of $\mathbf{B}$ consisting of the rows indexed by Ω, and $\bar{\mathbf{z}}_\Omega$ the subvector of $\bar{\mathbf{z}}$ consisting of the entries indexed by Ω.
Step 3: Solve the convex optimization (P_2)

$$(P_2) \qquad \mathbf{x}^\sharp = \mathrm{argmin}_{\mathbf{x}} \|\mathbf{x}\|_1, \quad \text{s.t. } \|\bar{\mathbf{z}}_\Omega - \mathbf{B}_\Omega \cdot \mathbf{x}\|_2 \le \epsilon.$$

Step 4: Return $\mathbf{y}^\sharp = \mathbf{B}_\Omega \cdot \mathbf{x}^\sharp$

In Algorithm 1, it is essential to ensure that the measurement matrix $\mathbf{B}_\Omega$ satisfies RIP even under hard thresholding. Actually, the thresholding process can be seen as a non-uniform Bernoulli sampling process where

$$\Pr[I_i = 1] = \begin{cases} 1 & \theta_i < \theta \\ 0 & \theta_i \ge \theta \end{cases}$$

Let $m = \mathbb{E}|\Omega|$. By employing Lemma 2.2 we know with probability exceeding $1 - 5e^{-c}\kappa_0^{n-m}\kappa_1^m$, $\mathbf{B}_\Omega$ satisfies RIP when $m \ge ck\log^4 n$. Here

$$\kappa_0 = \frac{n}{n-m}\ , \quad \kappa_1 = \frac{n}{m}$$

As m increases to n, κ decreases to 1, and $\mathbf{B}_\Omega$ satisfies RIP with higher probability. Lemma 2.2 gives a very strict bound for RIP. In practice, when the underlying data of interest and density distribution does not change drastically, the best threshold could be learned from historical data.

TR removes the noisiest measurements, and thus ensures a better bound on measurement error. Almost immediately, an improved error bound for final recovery can be formulated as below.

Theorem 4.1 (TR Performance Guarantee). *Let θ be a threshold in Algorithm 1 such that $\mathbf{B}_\Omega$ satisfies RIP with $\delta_{3k} + 3\delta_{4k} < 2$, where δ_k is the k-restricted isometry constraint of $\mathbf{B}_\Omega$. Then the output of Algorithm 1 satisfies*

$$\Pr\left[\|\mathbf{y}^\sharp - \mathbf{y}\|_2 \le C_k \cdot \epsilon_t\right] \ge (1 - \rho - \eta)^{|\Omega|}$$

where C_k is the constant in Lemma 2.3, $|\Omega|$ is the number of measurements left after the thresholding, and $\epsilon_t = \frac{\theta\sqrt{|\Omega|}}{\sqrt{\rho}}$.

Proof. Let $\bar{z}_\Omega(i)$ and $y_\Omega(i)$ be the i-th entries of the pruned sample vector $\bar{\mathbf{z}}_\Omega$ and signal $\mathbf{y}_\Omega$ in Algorithm 1. Since $\bar{z}_\Omega(i) \sim \mathcal{N}(y_\Omega(i), \frac{\sigma_i^2}{d_i})$, by Chebyshev's inequality,

$$\Pr\left((\bar{z}_\Omega(i) - y_\Omega(i))^2 \le \frac{\sigma_i^2}{d_i \rho}\right) \ge 1 - \rho. \tag{11}$$

By applying Eq. 6, 10 and union bound we have

$$\Pr\left((\bar{z}_\Omega(i) - y_\Omega(i))^2 \le \frac{\theta_i^2}{\rho}\right) \ge 1 - \rho - \eta. \tag{12}$$

For all $i \in \Omega$, $\theta_i \le \theta$, and therefore

$$\Pr\left((\bar{z}_\Omega(i) - y_\Omega(i))^2 \le \frac{\theta^2}{\rho}\right) \ge 1 - \rho - \eta. \tag{13}$$

Combine these for all $|\Omega|$ independent sample values:

$$\Pr\left(\|\bar{\mathbf{z}}_\Omega - \mathbf{y}_\Omega\|_2 \le \theta\sqrt{\frac{|\Omega|}{\rho}}\right) \ge (1 - \rho - \eta)^{|\Omega|} \tag{14}$$

Applying Lemma 2.3 with $\epsilon = \theta\sqrt{\frac{|\Omega|}{\rho}}$, we have

$$\Pr\left(\|\mathbf{y}^\sharp - \mathbf{y}\|_2 \le C_k \cdot \epsilon\right) \ge (1 - \rho - \eta)^{|\Omega|} \tag{15}$$

□

Compared with conventional CS, $\epsilon_t < \epsilon = \frac{\theta_{\max}\sqrt{n}}{\sqrt{\rho}}$, where $\theta_{\max} = \max_{1\le i\le n} \theta_i$ Therefore, the hard thresholding gives a tighter upper bound of the ℓ_2-norm error between $\bar{\mathbf{z}}_\Omega$ and $\mathbf{y}_\Omega$ than the original sample error $\bar{\mathbf{z}} - \mathbf{y}$. By applying conventional CS reconstruction algorithms like BP, we can reconstruct the signal and with high probability have this ℓ_2-norm error guarantee. As θ decreases, this reconstruction error bound improves. However, if θ is so small that the matrix $\mathbf{B}_\Omega$ does not satisfy the RIP requirements, the reconstruction is likely to fail. We confirm this logic experimentally in Section 6.2. Note that BP is not the only possible choice for the underlying recovery algorithm. Other algorithms, including OMP[26] and BCS[10] can also be plugged into Algorithm 1.

4.2 Weighted Recovery

Intuitively, Threshold Recovery can be viewed as a specific way of putting $\{0, 1\}$-weights on each measurement value to be ignored or fully trusted during recovery. In our proposed *Weighted Recovery* algorithm (WR), we generalize this and assign weights to each sampling to maximize the likelihood of the reconstructed signal. More specifically, WR assigns weights according to the measurement accuracy and reconstruct the signal under the constraints on weighted error. We show this is equivalent to maximize the likelihood of the reconstructed signal and the reconstructed error is bounded.

In conventional CS, we use a $\ell_1 - \ell_2$ *constraint optimization* to reconstruct the sparse signal.

$$(P_1)\ \ \min_{\mathbf{x}} \|\mathbf{x}\|_1 \ \text{ s.t. } \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \le \epsilon$$

The reconstruction can be viewed as a regression process, which consists of two parts: 1) minimize the ℓ_1 norm of the sparse representation, which ensures sparseness. 2) minimize the reconstruction error from the measurement, which ensures the reconstructed signal is close to the measurements. When the measurement errors are identically distributed, minimizing the ℓ_2 norm of $\mathbf{B}\mathbf{x} - \bar{\mathbf{z}}$ is equivalent to finding a reconstructed signal with maximum likelihood [22].

Algorithm 2 Weighted Recovery

Input: Observations $\{z_{i,j}\}_{1\le i\le n, 1\le j\le d_i}$, measurement matrix $\mathbf{B}_\Omega$, a threshold $\theta > 0$ and the weighted error constraint $\epsilon_w > 0$
Output: $\mathbf{y}^\sharp$, the estimation of original $\mathbf{y}$.

Step 1: Same as Step 1 in Algorithm 1.
Step 2: Solve the convex optimization (P_3)

$$(P_3) \qquad \begin{aligned} &\mathbf{x}^\sharp = \text{argmin}_{\mathbf{x}} \|\mathbf{x}\|_1, \\ &\text{s.t. } (\mathbf{B}_\Omega\mathbf{x} - \bar{\mathbf{z}})^T \boldsymbol{\Theta}^{-1} (\mathbf{B}_\Omega\mathbf{x} - \bar{\mathbf{z}}) \le \epsilon_w^2 \end{aligned}$$

Step 3: Return $\mathbf{y}^\sharp = \mathbf{B}_\Omega\mathbf{x}^\sharp$

However, sample errors vary wildly in our case, so we must take one step back: given the observations $\mathbf{Z}$, we would like to find a vector $\mathbf{x}$ such that 1) $\mathbf{x}$ is sparse enough 2) for a sample value $\bar{\mathbf{z}}$, the log-likelihood $\mathcal{L}(\mathbf{x}|\bar{\mathbf{z}})$ is maximized. Similar to (P1), we use an ℓ_1*-likelihood constraint optimization* to formalize the reconstruction goal, which is

$$\mathbf{x}^\sharp = \min_{\mathbf{x}} \|\mathbf{x}\|_1, \text{ s.t. } \mathcal{L}(\mathbf{x}|\bar{\mathbf{z}}) \ge \ell_0 \tag{16}$$

Here $\mathcal{L}(\mathbf{x}|\bar{\mathbf{z}})$ is the log-likelihood of a sparse representation $\mathbf{x}$ and ℓ_0 works as an adjustable parameter balancing sparsity and likelihood.

To compute $\mathcal{L}(\mathbf{x}|\bar{\mathbf{z}})$, we need to know the distribution of $\bar{\mathbf{z}}$. When d_i is large, $\bar{z_i}$ can be approximated by a normal distribution $\mathcal{N}(\bar{z_i}, \frac{\sigma_i^2}{d_i})$. However, when d_i is small, the distribution is elusive. We still employ the normal distribution to approximate, which is the default choice as in [18] when only the mean and the variation of a distribution are known according to the principle of maximum entropy. Therefore, in the computation, we use $\mathcal{N}(\bar{z_i}, \frac{\sigma_i^2}{d_i})$ to approximate the distribution of $\bar{z_i}$. The log-likelihood can be computed as

$$\mathcal{L}(\mathbf{x}|\bar{\mathbf{z}}) \propto -(\mathbf{B}_\Omega\mathbf{x} - \bar{\mathbf{z}})^T \boldsymbol{\Sigma}^{-1} (\mathbf{B}_\Omega\mathbf{x} - \bar{\mathbf{z}}) \tag{17}$$

where $\boldsymbol{\Sigma} = \mathbf{diag}(\frac{\sigma_1^2}{d_1}, \frac{\sigma_2^2}{d_2}, \ldots, \frac{\sigma_n^2}{d_i})$. This establishes that a lower bound ℓ_0 on the log-likelihood $\mathcal{L}(\mathbf{x}|\bar{\mathbf{z}})$ as in Eq. (16), is equivalent to an upper bound for the weighted measurement error as in Eq. (17). Let ϵ_w denote this *weighted error bound* in the following discussion.

Each $\frac{\sigma_i^2}{d_i}$ is unknown, but can be approximated by θ_i^2. Let $\boldsymbol{\Theta}$ be $\mathbf{diag}(\theta_1^2, \theta_2^2, \cdots, \theta_n^2)$. With this notation, we propose the Weighted Recovery algorithm in Algorithm 2. Notice that the weight of each sample $\bar{z_i}$ is θ_i^{-2}. Samples with higher accuracy are weighted higher in the reconstruction. Relative to Threshold Recovery, Weighted Recovery leverages all information available, while mitigating the potential risk of using inaccurate samples. WR is outlined in Algorithm 2, and its performance is guaranteed by the following theorem.

Theorem 4.2 (WR Performance Guarantee). *Let $\mathbf{B}_\Omega$ satisfies RIP with $\delta_{3k} + 3\delta_{4k} < 2$, where δ_k is the k-restricted isometry constant. For a signal $\mathbf{y} = \mathbf{B}_\Omega\mathbf{x}^\mathbf{0}$, with k-sparse $\mathbf{x}^\mathbf{0}$, the observations $\mathbf{z_i}$ for each y_i are i.i.d with $\mathbb{E}(z_i) = y_i$. Given a weighted error threshold ϵ_w, the output of Algorithm 2 satisfies*

$$\|\mathbf{x}^\sharp - \mathbf{x}^0\|_2 \le C_k \epsilon_w \theta_{\max}$$

with probability $(1 - \eta)^{2n}$. Here C_k is the constant from Lemma 2.3, η is a constant in Eq.(3), and $\theta_{\max}$ is $\max_i\{\theta_i\}$.

Proof. In Algorithm 2, ϵ_w is a fixed constant such that

$$(\mathbf{B}_\Omega \mathbf{x} - \bar{\mathbf{z}})^T \Sigma^{-1} (\mathbf{B}_\Omega \mathbf{x} - \bar{\mathbf{z}}) \leq \epsilon_w^2 \tag{18}$$

According to Eq. 10, we have

$$\Pr\left(\Sigma_{ii} = \frac{\sigma_i^2}{d_i} \leq \theta_i^2 = \Theta_{ii} \ \forall i = 1, \ldots, n\right) \geq (1-\eta)^n \tag{19}$$

As Σ and Θ are diagonal matrices with positive entries, it follows that with probability at least $(1-\eta)^n$,

$$(\mathbf{B}_\Omega \mathbf{x} - \bar{\mathbf{z}})^T \boldsymbol{\Theta}^{-1} (\mathbf{B}_\Omega \mathbf{x} - \bar{\mathbf{z}}) \leq \epsilon_w^2 \tag{20}$$

We only need to show that the "tube constraint" and the "cone constraint"[3] hold. For the "tube constraint", we want to show that the distance between the original sparse signal $\mathbf{B}_\Omega \mathbf{x}^\sharp$ and reconstructed signal $\mathbf{B}_\Omega \mathbf{x}^0$ is upper bounded.

$$(\mathbf{B}_\Omega \mathbf{x} - \bar{\mathbf{z}})^T \boldsymbol{\Theta}^{-1} (\mathbf{B}_\Omega \mathbf{x} - \bar{\mathbf{z}}) \leq \epsilon_w^2 \tag{21}$$

$$\Rightarrow \quad \sum_{i=1}^{|\Omega|} \frac{1}{\theta_i^2} (\mathbf{B}_{\Omega_i} \mathbf{x} - \bar{z}_i)^2 \leq \epsilon_w^2 \tag{22}$$

$$\Rightarrow \quad \|\mathbf{B}_\Omega \mathbf{x} - \bar{\mathbf{z}}\|_2^2 \leq \epsilon_w^2 \theta_{\max}^2 \tag{23}$$

And then, by the triangle inequality,

$$\|\mathbf{B}_\Omega \mathbf{x}^\sharp - \mathbf{B}_\Omega \mathbf{x}^0\|_2 \leq \|\mathbf{B}_\Omega \mathbf{x}^\sharp - \bar{\mathbf{z}}\|_2 + \|\bar{\mathbf{z}} - \mathbf{B}_\Omega \mathbf{x}^0\|_2 \tag{24}$$

Therefore, with probability at least $(1-\eta)^{2n}$,

$$\|\mathbf{B}_\Omega \mathbf{x}^\sharp - \mathbf{B}_\Omega \mathbf{x}^0\|_2 \leq 2\epsilon_w \theta_{\max} \tag{25}$$

The "cone constraint" holds following the proof of Theorem 1 in [3], by changing the error bound $\|\mathbf{B}_\Omega(\mathbf{x}^\sharp - \mathbf{x}^0)\|_2 = \|\mathbf{B}_\Omega \mathbf{h}\|_2$ from $2\epsilon_w$ to $2\epsilon_w \theta_{\max}$. Therefore, with probability at least $(1-\eta)^{2n}$,

$$\|\mathbf{x}^\sharp - \mathbf{x}_0\|_2 \leq C_k \cdot \epsilon_w \theta_{\max} \tag{26}$$

□

This theorem could be generalized to the cases where the samples are correlated as long as $\boldsymbol{\Theta}$ remains positive definite.

The RIP of the matrix $\mathbf{B}_\Omega$ is easy to satisfy in WR. Let $\Omega = \{i \mid d_i > 0\}$. Usually in a crowdsensing system, most samples have at least one observation. The measurement matrix $\mathbf{B}_\Omega$ in WR is the same as the one in conventional compressive sensing. It is reasonable to assume $|\Omega| = m \gg \Theta(k \log^4 n)$ and RIP is satisfied with high probability.

For practical use, we use θ_i^2 to approximate each $\frac{\sigma^2}{d_i}$, and by Eq.(6), the probability that there exists some $\theta_i^2 \leq \frac{\sigma^2}{d_i}$ such that the error of reconstruction is unbounded is at most $n\eta$. We also assume that $\bar{z_i}$'s are normally distributed. For each $\bar{z_i}$, when d_i, the number of observations, is large, observations of $\bar{z_i}$ can be approximately viewed as draws from $\mathcal{N}(y_i, \theta_i)$ by the Law of Large Numbers. On the other hand, when d_i is small, the distribution of $\bar{z_i}$ might be far from $\mathcal{N}(y_i, \theta_i)$, but θ_i is relatively large and lenient. So in both cases, assuming normality of $\bar{z_i}$'s has little influence on the weighted estimation error.

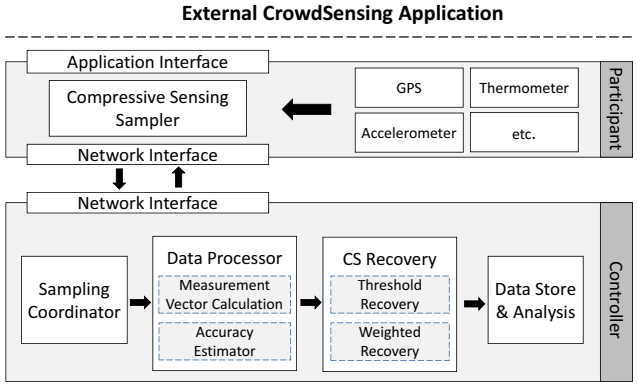


Figure 3: DACCS Implementation

5 SYSTEM IMPLEMENTATION

This section devotes to the design of our prototype implementation of DACCS. As depicted in Fig. 3, the current DACCS implementation consists of a participant component and a controller component integrating multiple functional modules of different purposes.

On the participant side, the "Compressive Sensing Sampler" incorporates with the central controller to trigger the sensing action of multiple "sensors" that are integrated or attached to the device. This procedure can also be done in an autonomous manner if the sampler is configured accordingly. The sensing data is directly measured by the sensors once they are triggered by the sampler. After deliver the sensing data back to the central controller via network interface, the participant finishes the loop of operations and waits until the next loop.

At the central controller side, the "sampling coordinator" works with each participants, and passes the received sensing data to the "data processor". Provided the original sensing data, the "data processor" calculates the corresponding measurement vectors, as well as the accuracy estimator as we introduced in the technical part, and passes them as the input data to the "CS Recovery" module which recovers the data of the whole sensing field and stores the recovered data the data storage module. The major modules are detailed as follows:

Sensors Operating as an event-driven Android services, upon request from the Compressive Sensing Sampler, a sensor replies with its probing results.

Compressive Sensing Sampler Compressive sensing sampler takes requests from the central controller, or autonomously dictates to trigger the action of sampling, based on signal strength, battery life, cost, time of the day, randomness, etc. A simplistic implementation could be sampling on a regular basis.

Sampling Coordinator Sampling coordinator is executed upon arrival of sampling data from participating mobile devices. It passes received data to "Data Processor" and also send back feedbacks to participants to further guide data sampling.

Data Processor Data Processor calculates measurement vectors from received data, and estimates the accuracy of the calculation which are further provided to the recovery module as the inputs.

CS Recovery The recovery module runs a DACCS algorithm – TR or WR – to recover the original data of interest by taking both

the sensed data and the corresponding accuracy estimates as input. The most time-consuming part is solving a convex optimization, for which polynomial-time algorithms exist and highly efficient software toolkits are available for our use.

Data Store and Analysis The data storage module keeps all the recovered data for further uses, including the operation of training parameters for DACCS algorithms.

6 EVALUATION

In this section, we first evaluate our algorithms using synthetic data to better understand their pros and cons, and the impact of corresponding parameters. We then use a real-life dataset to evaluate the effectiveness of our algorithms in practice.

6.1 Methodology

In all evaluations, we use the ℓ_2-norm error $\|\mathbf{y}^\sharp - \mathbf{y}\|_2$ as the measure of accuracy. For comparison, we consider the following baselines:

- **Basis Pursuit [6] (BP):** standard ℓ_1-norm minimization algorithm solving (P_1), run naïvely on the sample means.
- **Bayesian Compressive Sensing [10] (BCS):** an iterative algorithm optimizing a sparsity-based probabilistic graphical model, running directly on the sample means. This framework considers Gaussian process priors over the compressed sparse signal space and has the capability to propagate uncertainties to the output measurement space by considering the constraints imposed by the projection matrix. This is one of the most cited works for compressible signal recovery with Gaussian noise.
- **TR with ground truth (TR(G)):** TR with known measurement error. Therefore, measurements with the largest error are discarded, which maximizes the effectiveness of removing noise. This is an unrealistic algorithm only included as a performance lower bound on TR.
- **CS-UTS [33]:** a compressive sensing approach to monitoring urban traffic, which is the same application scenario as our practical evaluation.

Parameters in our algorithms and baselines are optimized in the evaluation.

6.2 Numerical Results

First, we study the case where each $\mathbf{x}$ is a *perfectly sparse vector*, i.e., $\mathbf{y}$ is perfectly sparse under the Fourier basis $\mathbf{B}$ ($\mathbf{y} = \mathbf{Bx}$). Specifically, we construct random 5-sparse vectors $\mathbf{x}$ of dimension 512 (i.e., $n = 512, k = 5$). Note that the length n and sparsity k only affect the minimum number of entries necessary for successful recovery, per the factor $k \log n$ as introduced in Sec. 2. Thus this choice of $n = 512, k = 5$ can effectively evaluate our algorithms under representative sparse signals.

We draw *observations* $z_{i,j}$ for each entry s_i independently, where $z_{i,j} \sim \mathcal{N}(s_i, \sigma_i^2)$ for $j = 1, 2, \cdots, d_i$. We uniformly select each σ_i ahead of time from $[0, 0.5]$ to represent the *observations' accuracy*. The upper bound $\sigma_{max} = 0.5$ is of the same order as y_i, i.e., samples are very noisy.

We observe similar results when the observations are drawn from distributions other than the normal distribution, given appropriately chosen parameters. Therefore we use this normal distribution-based setup throughout the numerical experiment section.

6.2.1 Performance. We compare the performance of TR/WR in the case of perfectly sparse signals in Fig. 4. In each test, over all $n = 512$ entries, we make 51200 observations (for an average of 100 observations per sample). We consider normally, uniformly, and exponentially distributed observation counts. Intuitively, observation count distributions simulate the number of observations for each mean, and hence how confident we are about the mean. Normally distribution results in mostly average confidence and a few very confident or very skeptical on means; uniform indicates a wide range ofconfidence on means; and exponential shows mostly low confidence and few high confidence.

From the figure we can see that in all three situations, the reconstruction error of TR/WR in ℓ_2-norm is significantly lower than that of all baselines. To be specific, TR/WR reduces error by more than 70% in all cases. In the normally-distributed case which favors the baselines most, the baselines have more than 4× the error of ours. In the exponentially-distributed case the ℓ_2-norm errors for TR/WR are only 10% of the baselines' errors. Moreover, TR/WR is much less sensitive to how the observation counts are distributed – they have similar performance in all three cases, while the baselines clearly suffer from larger errors in the uniformly- and exponentially-distributed cases.

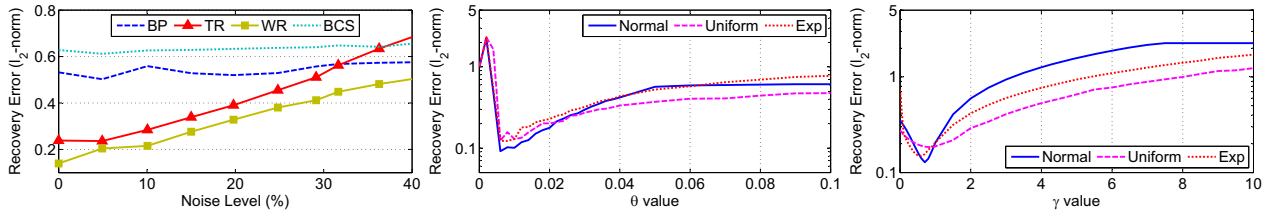
6.2.2 Robustness. In practice, most signals are not perfectly sparse, i.e. $\mathbf{x}$ may consist of a few major coefficients with significant magnitudes and a number of insignificant (but nonzero) ones (such as the traffic dataset we evaluate in the next section). In Fig. 4(a), we present how different level of non-sparsity in $\mathbf{x}$ affects performance. Here, the non-sparsity level (%) is measured in terms of the ratio between noise ℓ_2-norm and $\|\mathbf{x}\|_2$.

As noise increases, we observe a gradual increase in recovery error of both TR and WR - our algorithms are more sensitive to non-sparsity in $\mathbf{x}$ compared to baselines. Despite of it, with reasonable noise level (0% to 30%), TR/WR still outperform the baselines.

The key parameter in TR is the threshold θ that determines how many samples are discarded, which is evaluated in Fig. 4(b). The three curves correspond to the uniformly-, normally- and exponentially-distributed cases as introduced before. As θ increases, less samples are discarded. In all three cases, we see clear and similar improvement as we discard samples with big errors. The optimal θ for each case is about 0.02. If we reduce θ – discarding more – a steep increase in recovery error appears because of violation of the RIP constraints, i.e. we discard too many samples. Fig. 4(c) depicts the impact of the parameter γ in WR. Similarly, the optimal parameter is $\gamma \approx 0.8$ for all three cases. In practice, when the distribution of the data is unknown but stable over time, we found it was effective to *train* a good value for γ on historical data and subsequently use this value to reconstruct new data.

6.3 Practical Scenario: Traffic Monitoring

Traffic monitoring is a prominent example in crowdsensing, where the crowdsensing system aims to monitor the road conditions of



(a) Non-sparcity in vector $\mathbf{x}$ (b) Threshold θ in TR (c) Parameter γ in WR

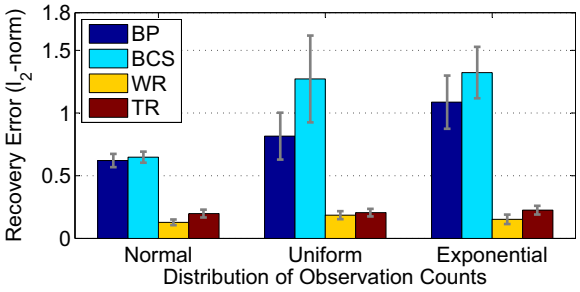


Figure 4: Performance comparison in numerical experiments

main road segments by gathering driving speed reports from participants, i.e. cars in the city, and estimates the average velocity for each target road segment. Previous work has successfully built up CS-based crowdsensing systems for this practical scenario [33], where CS is used to greatly reduce the total number of necessary samples. Furthermore, this scenario also demonstrates how the density affects the effectiveness of CS-based methods.

Following the setup in [33], we divide the arterial roads of Beijing into road segments of 200m, as shown in Fig 5(a). As in Fig. 5(b) and 5(c), average driving speed shows temporal and spatial correlation. Instead of monitoring all the road segments, we pick the busiest 389 road segments to monitor, every 30 minutes during work hours for 3 continuous weeks.

Using all available reports of $\approx$ 10000 taxis from the open dataset[4], we calculate the average driving speed as the ground truth $\mathbf{y}$, which is a vector containing 389 elements as shown in Fig. 5(d). Furthermore, we employ the DCT-II matrix as the sparse basis $\mathbf{B}$ since the time series $\mathbf{y}$ is sparse in the frequency spectrum, as shown in Fig. 5(e). Again, samples are uniformly and randomly selected and ℓ_2-norm error is used as the accuracy metric.

6.3.1 Dataset and Parameter Setup. Based on the traffic data, we observe that 1) during regular traffic (say 11AM), the driving speed distribution is approximately normal, with speeds depending on the road condition; and 2) during traffic congestion (say 6:30pm), the distribution is approximately exponential. Furthermore, the distribution of the number of observations over all 389 target entries is approximately Gaussian with a mean of $\approx$ 120.

As discussed in Section 6.2.2, the optimal value of θ in TR and γ in WR could be trained from historical data. Fig. 5(f) provides justification for choosing the parameter γ from historical data. Fixing a sampling rate, over 500 runs we generate a histogram of the ℓ_2-norm error-minimizing value of γ for random observation selections from historical data. The curve plots γ versus recovery error for the test data, averaged over 50 selections. Most of the mass of the histogram is near the average optimal value of γ, which is $\approx$ 1800, suggesting that training this parameter on past data results in a successful recovery. The optimal θ in TR is obtained similarly from historical data.

6.3.2 Performance. Fig. 6 depicts the comparison between the recovery performance versus sampling rate for all the algorithms. We see that TR and WR generally outperform BP and BCS regardless of sampling rate. Even when the sampling rate is extremely low, TR and WR perform particularly well, e.g. TR/WR reduce ℓ_2-norm error 20%/40% over BP, and 40%/50% over BCS respectively at a sampling rate of 10%. TR/WR also compare well to other state-of-the-art methods like an adapted version of the algorithm from [33]; at the same sampling rate of 10%, TR/WR reduce ℓ_2-norm error 40%/50% over this method, called CS-UTS.

Furthermore, we compare TR with one special baseline – TR with the ground truth (TR(G)), i.e. perfect knowledge of how noisy each term is. TR performs very closely to this ideal case when sampling rates exceed 30%. Moreover, WR outperforms this baseline most of the time.

Last, as sampling rate increases, all algorithms perform well because enough observations are collected to generate accurate sample values.

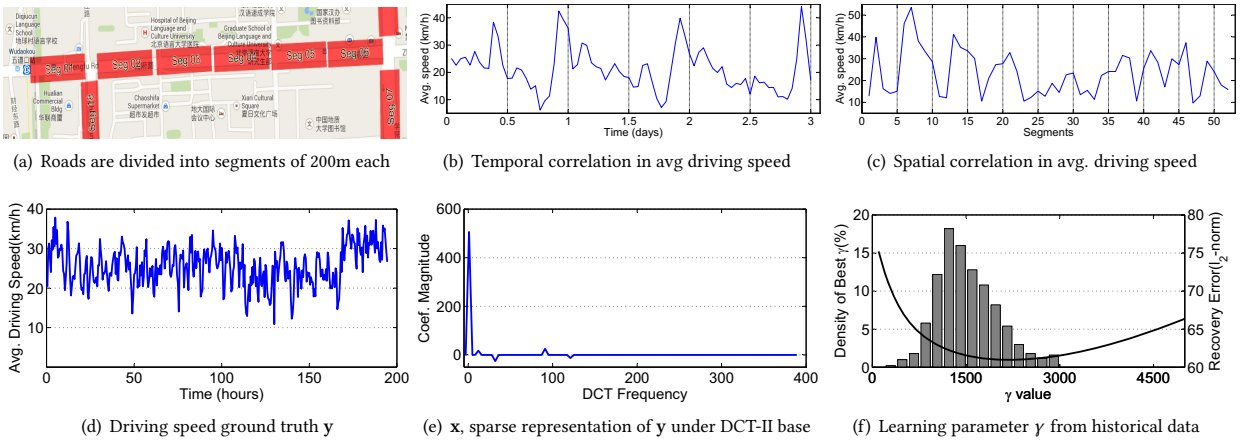
7 RELATED WORK

7.1 Compressive Sensing

CS is a powerful technique for sparse approximation, and has been a hot topic in multiple fields including approximation algorithms, information theory, signal processing and networked systems (e.g. [4, 6, 15, 25]). The major feature of CS is (exact) recovery of signals via only a small number – much lower than the Nyquist limit – of random samples or measurements. As long as the underlying signal is sparse, recovery is feasible without any further knowledge, which makes it particularly suitable for practical use. Most relevant to our work is the establishment of RIP requirement for matrices of various structures. [4] first proposed RIP for Gaussian random matrices, and similar results were formulated for random partial orthogonal systems with $O(k \log^6 n)$ factor. Further this factor was improved to $O(k \log^4 n)$[7]. Besides, [25] provided RIP for matrices that are constructed at non-uniformly random.

Another main challenge in applying CS is processing noisy measurements. Candès et. al. proposed an l_1-regularization method which has provable recovery guarantees [3]. This result was later

[4]Traffic monitoring data: http://sensor.ee.tsinghua.edu.cn/



(a) Roads are divided into segments of 200m each (b) Temporal correlation in avg driving speed (c) Spatial correlation in avg. driving speed

(d) Driving speed ground truth $\mathbf{y}$ (e) $\mathbf{x}$, sparse representation of $\mathbf{y}$ under DCT-II base (f) Learning parameter γ from historical data

Figure 5: Traffic dataset

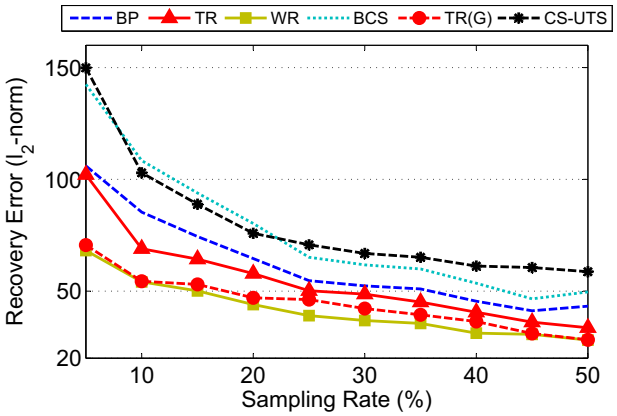


Figure 6: Performance comparison in traffic monitoring scenario

generalized to exponential family noise by Rish and Grabarnik [22]. There are results which work for specific types of noises, e.g. Poisson noise was considered by Raginsky et. al. [20] in the context of photon limited imaging in optical systems; and Bayesian CS [10] can only work with i.i.d. Gaussian noises. New CS recovery algorithms were developed for two-component noise model[17] and measurements with sparse noise[14]. These results do not directly fit our scenario where noises are more general.

7.2 Mobile Crowdsensing

Crowdsensing has become a promising means of data acquirement. Mainstream smartphones and recently popular wearable devices like Google Glass and Narrative Clip are equipped with many sensors. These devices can be used as abundant sources of raw data. By aggregating this data, a number of crowd-based systems have been built for applications like trajectory monitoring [12], traffic monitoring [29, 33], location characterization [5], and building noise maps [21, 30]. Within a wider range of mobile networking, [15] first explored the utilization of CS in the community of mobile networks, and further extended by [28]. Recently, [19] proposed a CS-based method to estimate congestion in urban traffic via moving probe data in cars.

8 CONCLUSION

In this paper we explore density disparity in crowdsensing systems where observations are noisy and of different density. In order to better estimate the sample error when we average these observations, we formulate a probabilistic upper bound on the variance of observations. By discarding samples according to a threshold based on this variance bound, we improve the recovery accuracy of CS schemes. Further, we generalize this discarding algorithm (TR) into a more general weighted algorithm (WR), which employs the variance upper bounds as weights to maximize the recovery likelihood. We prove the performance guarantees for TR and WR. Using both synthetic and real-life data traces, TR/WR are shown to outperform conventional CS recovery algorithms.

TR/WR is our first attempt in studying density disparity, which has a great potential in CS recovery algorithm improvement and many other application scenarios. In this paper we used only BP in implementing TR/WR algorithms, we believe that other CS recovery algorithms (like matching pursuit based algorithms) and even probabilistic CS framework (like BCS) could benefit from the idea of TR and WR. There is also much room for future research, such as recovering statistics other than population mean, considering different observation costs in the sampling stage, and other practical issues.

REFERENCES

[1] Richard Becker, Ramón Cáceres, Karrie Hanson, Sibren Isaacman, Ji Meng Loh, Margaret Martonosi, James Rowland, Simon Urbanek, Alexander Varshavsky, and Chris Volinsky. 2013. Human Mobility Characterization from Cellular Network Data. *Commun. ACM* 56, 1 (Jan. 2013), 74–82. DOI:http://dx.doi.org/10.1145/2398356.2398375

[2] Emmanuel J Candès. 2006. Compressive sampling. In *Proceedings of the International Congress of Mathematicians*. 1433–1452.

[3] Emmanuel J Candès, Justin K Romberg, and Terence Tao. 2006. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics* 59, 8 (2006), 1207–1223.

[4] Emmanuel J Candès and Terence Tao. 2006. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on* 52, 12 (2006), 5406–5425.

[5] Yohan Chon, Nicholas D. Lane, Fan Li, Hojung Cha, and Feng Zhao. 2012. Automatically characterizing places with opportunistic crowdsensing using smartphones. In *UbiComp '12*. ACM, 481–490.

[6] David L Donoho. 2006. Compressed sensing. *Information Theory, IEEE Transactions on* 52, 4 (2006), 1289–1306.

[7] Simon Foucart and Holger Rauhut. 2013. *A mathematical introduction to compressive sensing*. Springer.

[8] R.K. Ganti, Fan Ye, and Hui Lei. 2011. Mobile crowdsensing: current state and future challenges. *Communications Magazine, IEEE* 49, 11 (2011), 32–39.

[9] Puneet Jain, Justin Manweiler, Arup Acharya, and Kirk Beaty. 2013. FOCUS: Clustering Crowdsourced Videos by Line-of-sight. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)*. ACM, New York, NY, USA, Article 8, 14 pages. DOI: http://dx.doi.org/10.1145/2517351.2517356

[10] Shihao Ji, Ya Xue, and Lawrence Carin. 2008. Bayesian compressive sensing. *Signal Processing, IEEE Transactions on* 56, 6 (2008), 2346–2356.

[11] John Francis Kenney and Ernest Sydney Keeping. 1954. *Mathematics of Statistics-Part One*. D. Van Nostrand Company Inc.

[12] Linghe Kong, Liang He, Xiao-Yang Liu, Yu Gu, Min-You Wu, and Xue Liu. 2015. Privacy-preserving compressive sensing for crowdsensing based trajectory recovery. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*. IEEE, 31–40.

[13] Nicholas D Lane, Yohan Chon, Lin Zhou, Yongzhe Zhang, Fan Li, Dongwon Kim, Guanzhong Ding, Feng Zhao, and Hojung Cha. 2013. Piggyback CrowdSensing (PCS): energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 7.

[14] Jason N Laska, Mark A Davenport, and Richard G Baraniuk. 2009. Exact signal recovery from sparsely corrupted measurements through the pursuit of justice. In *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference on*. IEEE, 1556–1560.

[15] Chong Luo, Feng Wu, Jun Sun, and Chang Wen Chen. 2009. Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 145–156.

[16] Michael Lustig, David Donoho, and John M Pauly. 2007. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic resonance in medicine* 58, 6 (2007), 1182–1195.

[17] Brian Moore and Balasubramaniam Natarajan. 2015. A compressive sensing based analysis of anomalies in generalized linear models. *Communications in Statistics-Theory and Methods* 44, 13 (2015), 2705–2719.

[18] Sung Y Park and Anil K Bera. 2009. Maximum entropy autoregressive conditional heteroskedasticity model. *Journal of Econometrics* 150, 2 (2009), 219–230.

[19] Xiao Qi, Yongcai Wang, Yuexuan Wang, and Liwen Xu. 2014. Compressive Sensing over Strongly Connected Digraph and Its Application in Traffic Monitoring. In *INFOCOM 2014, IEEE*.

[20] Maxim Raginsky, Rebecca M Willett, Zachary T Harmany, and Roummel F Marcia. 2010. Compressed sensing performance bounds under Poisson noise. *Signal Processing, IEEE Transactions on* 58, 8 (2010), 3990–4002.

[21] Rajib Rana, Chun Tung Chou, Salil Kanhere, Nirupama Bulusu, and Wen Hu. 2010. Ear-Phone: An End-to-End Participatory Urban Noise Mapping. In *IPSN '10*. ACM.

[22] Irina Rish and Genady Grabarnik. 2014. Sparse signal recovery with exponential-family noise. In *Compressed Sensing & Sparse Filtering*. Springer, 77–93.

[23] Darshan Santani, Jidraph Njuguna, Tierra Bills, Aisha W. Bryant, Reginald Bryant, Jonathan Ledgard, and Daniel Gatica-Perez. 2015. CommuniSense: Crowdsourcing Road Hazards in Nairobi. In *MobileHCI '15*. ACM, New York, NY, USA, 445–456. DOI: http://dx.doi.org/10.1145/2785830.2785837

[24] Rijurekha Sen, Youngki Lee, Kasthuri Jayarajah, Archan Misra, and Rajesh Krishna Balan. 2014. GruMon: Fast and Accurate Group Monitoring for Heterogeneous Urban Spaces. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems (SenSys '14)*. ACM, New York, NY, USA, 46–60. DOI: http://dx.doi.org/10.1145/2668332.2668340

[25] Yiran Shen, Wen Hu, Rajib Rana, and Chun Tung Chou. 2013. Nonuniform compressive sensing for heterogeneous wireless sensor networks. *Sensors Journal, IEEE* 13, 6 (2013), 2120–2128.

[26] Joel A Tropp and Anna C Gilbert. 2007. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on* 53, 12 (2007), 4655–4666.

[27] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No Need to War-drive: Unsupervised Indoor Localization. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services (MobiSys '12)*. ACM, New York, NY, USA, 197–210. DOI: http://dx.doi.org/10.1145/2307636.2307655

[28] Xiaopei Wu and Mingyan Liu. 2012. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 1–12.

[29] Liwen Xu, Xiaohong Hao, Nicholas D Lane, Xin Liu, and Thomas Moscibroda. 2015. Cost-aware compressive sensing for networked sensing systems. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*. ACM, 130–141.

[30] Liwen Xu, Xiaohong Hao, Nicholas D Lane, Xin Liu, and Thomas Moscibroda. 2015. More with less: Lowering user burden in mobile crowdsourcing through compressive sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 659–670.

[31] Dejun Yang, Guoliang Xue, Xi Fang, and Jian Tang. 2012. Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing. In *Mobicom '12*. ACM, 173–184.

[32] Yin Zhang, Matthew Roughan, Walter Willinger, and Lili Qiu. 2009. Spatio-temporal compressive sensing and internet traffic matrices. In *ACM SIGCOMM Computer Communication Review*, Vol. 39. ACM, 267–278.

[33] Yanmin Zhu, Zhi Li, Hongzi Zhu, Minglu Li, and Qian Zhang. 2013. A Compressive Sensing Approach to Urban Traffic Estimation with Probe Vehicles. *Mobile Computing, IEEE Transactions on* 12, 11 (2013), 2289–2302. DOI: http://dx.doi.org/10.1109/TMC.2012.205