

Cost-Aware Compressive Sensing for Networked Sensing Systems

Liwen Xu[†], Xiaohong Hao[†], Nicholas D. Lane[‡], Xin Liu[‡], Thomas Moscibroda[‡]
[†]Tsinghua University, [‡]Microsoft Research, [‡]U.C. Davis

ABSTRACT

Compressive Sensing is a technique that can help reduce the sampling rate of sensing tasks. In mobile crowdsensing applications or wireless sensor networks, the resource burden of collecting samples is often a major concern. Therefore, compressive sensing is a promising approach in such scenarios. An implicit assumption underlying compressive sensing – both in theory and its applications – is that every sample has the same cost: its goal is to simply reduce the *number of samples* while achieving a good recovery accuracy. In many networked sensing systems, however, the cost of obtaining a specific sample may depend highly on the location, time, condition of the device, and many other factors of the sample.

In this paper, we study compressive sensing in situations where different samples have different costs, and we seek to find a good trade-off between minimizing the total sample cost and the resulting recovery accuracy. We design Cost-Aware Compressive Sensing (CACS), which incorporates the cost-diversity of samples into the compressive sensing framework, and we apply CACS in networked sensing systems. Technically, we use regularized column sum (RCS) as a predictive metric for recovery accuracy, and use this metric to design an optimization algorithm for finding a least cost randomized sampling scheme with provable recovery bounds. We also show how CACS can be applied in a distributed context. Using traffic monitoring and air pollution as concrete application examples, we evaluate CACS based on large-scale real-life traces. Our results show that CACS achieves significant cost savings, outperforming natural baselines (greedy and random sampling) by up to 4x.

Categories and Subject Descriptors

H.4 [Information Systems Applications]: Miscellaneous

General Terms

Design, Experimentation, Performance.

Keywords

Crowdsensing, Compressive Sensing, Resource-efficiency.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

IPSN '15, April 14 - 16, 2015, Seattle, WA, USA.

Copyright 2015 ACM 978-1-4503-3475-4/15/04 ...\$15.00.

10.1145/2737095.2737105.

1. INTRODUCTION

Fueled by the ever increasing sophistication and diversity of sensors, there has been a significant growth and momentum of networked sensing systems based on mobile devices and sensor networks in recent years. Example applications of such systems include traffic and road condition monitoring, noise and ambiance watch, air quality and pollution monitoring and social computing consumption. It is well-known that in such large-scale systems, the sensor resource – i.e. energy cost, computing power, and bandwidth consumption – imposes a major obstacle. For example, most sensing applications require location information and GPS is known for its high power consumption. Therefore, such resource burden often hinders the necessary participation and widescale adaption of the targeting applications.

Thus, *sampling* is the natural method of choice for such networked sensing scenarios. The goal of sampling is to sample (i.e., collect, upload, and process) only a small subset of the totally available data, and to then reconstruct the underlying data or the desired aggregate information based on this subset. In this context, a particularly promising approach is Compressive Sensing (CS). The idea of compressive sensing is to exploit the inherent sparsity and data redundancy in these application scenarios in order to reduce the sampling rate. Such data redundancy is indeed common and manifests itself in numerous ways. For example, the traffic condition in a given area/city typically exhibits strong temporal and spatial correlation [33]. By leveraging this inherent correlation, CS-based sampling and recovery techniques have the potential to significantly lower the number of samples required, and thus reduce the burden on the mobile devices.

One implicit assumption in the traditional compressive sensing framework is that every sample has equal cost. The goal of compressive sensing has been to reduce the *number of samples* needed to achieve the desired recovery accuracy; and therefore the question of *which samples* to take has been viewed only with regard to its impact on the recovery performance. This assumption that all sample costs are equal is not surprising. Compressive sensing has been widely used in fields such as image compression, medical imaging, and geophysical data analysis in which indeed there is no difference between the cost of one sample or another. Besides, the mathematical foundation of compressive sensing is also built upon the assumption that sample costs are identical.

In practical networked sensing systems, however, there exists significant *sample cost diversity*, and this diversity should be integrated into the compressive sensing framework. The reasons for sample cost diversity are manifold. Sensors are inherently diverse, and their conditions are time-varying and location-dependent. For example, the energy consumption

of obtaining a GPS lock depends on many factors, including location, atmosphere, and hardware conditions. The cost of reporting a sample to a central server varies based on the network condition, cellular data plan, distance to the nearest cell tower, or other concurrent activities on the device. In addition, in a mobile sensing system, the cost may also depend on users' perception: e.g. a user may not mind contributing a sample normally, but would consider the sample expensive when he is running out of battery. In short, different samples can have vastly different costs in networked sensing systems; and in order to leverage its potential benefits, we must effectively incorporate cost-diversity into the compressive sensing framework.

In this paper, we design a principled resource-efficient solution to networked sensing systems based on *Cost-Aware Compressive Sensing* (CACS). Our approach integrates *cost diversity* into compressive sensing, and seeks to find a good balance between the total sampling cost and the resulting recovery accuracy. Doing so is challenging. For one, the need to select samples who collectively have low total cost fundamentally runs counter to the need to capture the sparse structure no matter in randomized or deterministic ways in CS. More generally, there is a trade-off between choosing low-cost samples and achieving a sufficiently wide-spread and balanced coverage of the entire signal space. For example, a naive greedy approach that always selects samples with low cost will necessarily result in a poor recovery of the collected data: If GPS samples in a downtown area with highrise buildings are expensive; the greedy algorithm will avoid such samples all together and lose sufficient coverage of the downtown area. Thus, the fundamental challenge is to find a set of samples that are low-cost, and yet capture the underlying data's complex structure to allow for accurate recovery.

The key technical ingredient of our approach is to formulate *Regularized Column Sum* (RCS) as a practical metric for predicting recovery accuracy. We prove that RCS provides theoretical recovery guarantees while being easily computable. This is in contrast to existing compressive sensing work, in which provable recovery lower bounds have been based on the sensing matrix and the so-called Restricted Isometry Property (RIP). This classical approach is not practical in our setting, because i) it is NP-hard to verify the RIP condition of a sampling matrix, and ii) RIP only provides an insufficiently loose lower bound on recovery accuracy. The key is that RCS allows us to devise a novel convex optimization-based approach (RCS-constrained Optimization) for finding the least cost randomized sampling scheme using relaxation. This can be done efficiently, and it can also be analyzed in terms of its performance.

Finally, we present two methods to apply the Cost-Aware Compressive Sensing framework in decentralized systems. The two heuristics – Distributed Weighted Sampling and Pairwise Sampling – are complementary in their approach to cost-awareness. We evaluate our techniques using real-life traces on air quality monitoring and traffic monitoring, collected from a large metropolitan area over the course of several weeks. Our results show that Cost-Aware Compressive Sensing methods result in significant cost savings for the same recovery accuracy. For example, they substantially outperform the standard baseline algorithms (the naive greedy algorithm by up to 4x; the classic cost-oblivious sampling algorithm by 3x) in a large variety of settings.

2. COMPRESSIVE SENSING PRIMER

Compressive Sensing (CS), a recent breakthrough in the signal processing community, is an efficient technique of sampling high dimensional data that has an underlying *sparse structure*. In particular, it is possible to *sample* at a rate much lower than the Nyquist sampling rate, and then nevertheless accurately *reconstruct* signals via a linear projection in subspace. Generally speaking, CS first measures a small group of linear projections of the target data, and then *reconstructs* the original data via the incomplete information. CS is particularly promising in practical applications that exhibit spatiotemporal correlation and data redundancy.

Sparse Structure: Consider a target vector $\mathbf{y} \in \mathbb{R}^n$, and let \mathbf{y} be decomposed under a certain base Ψ , i.e. $\mathbf{y} = \Psi\mathbf{x}$ where \mathbf{x} is the coefficient vector. \mathbf{x} is called *k-sparse* if it has only k non-zero entries. Discrete Fourier base and discrete cosine base are examples of typical choices for the sparsifying base Ψ .

Formally, assuming Φ is a linear encoder which projects an n -dimensional data into an m -dimensional subspace ($m < n$), CS can accurately reconstruct \mathbf{y} that has such sparse structure from its linear measurements

$$\mathbf{s} = \Phi\mathbf{y} = \Phi\Psi\mathbf{x} \quad (1)$$

even if m is as small as $O(k \log \frac{n}{k})$ [6]. Note that in order for the recovery to be successful, $\Phi\Psi$ has to satisfy the so-called Restricted Isometry Property (RIP) [7]. Here Φ , Ψ and $\mathbf{A} = \Phi\Psi$ are called sampling matrix, sparsifying matrix and sensing matrix, respectively.

The RIP constant of a matrix \mathbf{A} is defined as the smallest positive δ_k that satisfies $1 - \delta_k \leq \frac{\|\mathbf{A}\mathbf{v}\|_2}{\|\mathbf{v}\|_2} \leq 1 + \delta_k$ for all k -sparse vectors \mathbf{v} . A small δ_k is the key to guarantee the success of CS reconstruction. Importantly, randomly structured \mathbf{A} meet this requirement with high probability [6].

Intuitively, various types of signals in real-life, such as temperature values [18], soil moisture [29] and traffic condition on road networks [33] should be spatially correlated, and also have high correlation and periodicity over time. These signals usually inhere sparse structures, but not perfectly sparse. Instead, they are called *compressible* because their coefficients under the sparsifying base decay at exponential rate when sorted by magnitude. Similar results also hold for compressible signals in CS theory. Natural and compressible signals are the main focus in this paper.

Random Sampling: Among the various possible designs of the linear encoder Φ , the *random partial identity matrix*, which implies *random sampling*, is one basic and popular choice because of its stability in performance and simplicity in practice. By applying an $m \times n$ random partial identity matrix in (1), \mathbf{s} is simply a random sample of \mathbf{y} of size m .

Data Reconstruction: As for data reconstruction, CS performs the ℓ_1 -norm minimization (ℓ_1 -min)

$$(P_1) \arg \min_{\hat{\mathbf{x}} \in \mathbb{R}^n} \|\hat{\mathbf{x}}\|_{\ell_1}, \text{ subject to } \|\Phi\Psi\hat{\mathbf{x}} - \mathbf{y}\|_{\ell_2} \leq \epsilon.$$

This replaces the NP-hard ℓ_0 -norm minimization that directly searches for the sparsest $\hat{\mathbf{x}}$. ℓ_1 -min can be solved in polynomial-time by linear programming, and has nice guarantees in terms of recovery error [7]. Besides, various greedy algorithms are also practical alternatives for ℓ_1 -min, e.g. Compressive Sampling Matching Pursuit (CoSaMP) [19].

3. Cost-Aware Compressive Sensing

Conventional CS makes the implicit assumption that all samples have the same cost. This assumption is mostly valid in existing applications domains of CS (e.g. image recovery), however it is different in the context of networked sensing systems, where the cost of taking and reporting samples can vary significantly.

In the following, we first discuss the various types of costs occurring in networked systems. Then we present the challenges in incorporating costs into compressive sensing, which motivate our effort in designing *Cost-Aware Compressive Sensing* (CACS) for resource-efficient solutions in such cases. The challenges in CACS mainly arise from two aspects: i) the difficulty of balancing between recovery accuracy and sampling cost, and ii) predicting recovery accuracy for a given sampling strategy.

3.1 Cost Factors

In practice, different types of costs can occur in wireless networks, including but not limited to 1) energy consumption; 2) data consumption; and 3) perception cost.

Energy Consumption. Because of limited battery budget, energy consumption is critical in devices like sensor nodes and mobile phones. Devices consume energy in both measuring and reporting a sample (e.g. locate a GPS signal and report position). This cost depends on the location as well as the status of the device. The reporting cost may depend on the network (i.e., WiFi, 2G, 3G or 4G), the signal strength, variability to the network, and the congestion level.

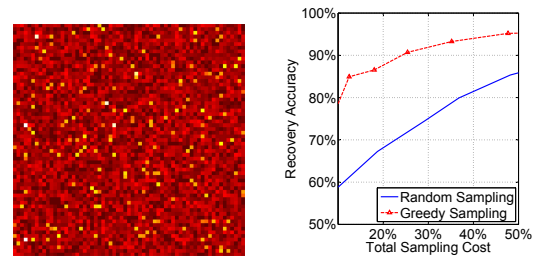
Data Consumption. Meanwhile, the reporting may incur cellular data cost when using cellular networks. As discussed in [30], a major burden for *emerging* crowdsensing applications may be network bandwidth.

Perception Cost. Finally, users may have different perception of a given cost. For example, a user with a cellphone with full battery may not consider the energy consumption for GPS locating to be high, whereas other users may be more sensitive to the same amount of energy usage. Such perception-based cost adjustments should be considered as they are important to user experience.

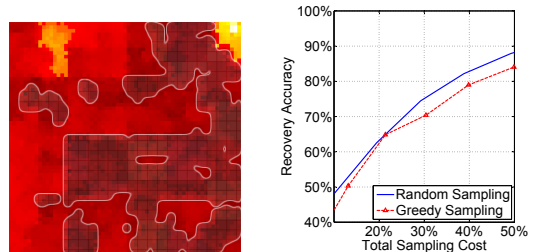
Spatial temporal Correlation. It is important to observe that sampling costs are often spatially and temporally correlated among devices/users. E.g. in an area with poor cellular coverage, all users are likely to incur high transmission cost. Furthermore, the battery status of users would show high temporal correlation and more users are likely to have low remaining batteries towards the end of the day. Such correlations in general render naive cost-aware sampling methods less effective as discussed next.

3.2 Challenge: Balancing Accuracy and Cost

According to CS theory, samples picked uniformly at random will meet the requirement of high recovery accuracy with high probability. Although randomness is not the only way to ensure accurate recovery, *random sampling* is employed most often in previous work because of its simplicity and robustness in practice. On the other hand, to reduce sampling cost, it is necessary to avoid the samples that are more expensive, and the uniform randomness is inevitably violated by favoring samples with lower cost. Naturally, we would like to find a balance between these two opposing forces – the need for keeping high recovery accuracy and the desire of choosing lower cost samples. Thus, Cost-Aware



(a) On i.i.d cost map, greedy strategy works well



(b) On spatially-correlated cost map, greedy fails

Figure 1: Motivation of CACS

Compressive Sensing(CACS) pursues the two-fold objective of:

$$\begin{cases} \max & \text{Recovery accuracy;} \\ \min & \text{Sampling cost.} \end{cases}$$

CACS is a randomized sampling process. To clarify the subsequent discussion, we first introduce the notations and definitions in CACS. In a networked system containing n nodes, $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ is the signal of interest that are separately held by different network nodes. \mathbf{y} is compressible, and has a sparse representation under the base Ψ , i.e. $\mathbf{y} = \Psi\mathbf{x}$, in our case Ψ is the Fourier base. Sampling costs differ from node to node (or rather, from sample to sample), and $\mathbf{c} = \{c_1, c_2, \dots, c_n\}$ denotes the corresponding costs at different nodes.

The CACS sampling process is conducted independently by each node flipping a coin according to $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$, $0 \leq \pi_i \leq 1$ which are the probabilities assigned to the nodes. Use $\Omega := \{i | X_i = 1, X_i \sim \text{Bernoulli}(1, \pi_i)\}$ to denote the index set of a sample instance, and let $m = \mathbb{E}|\Omega| = \sum \pi_i$ be the expected sample size. Let $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ denote the recovery of \mathbf{x} and \mathbf{y} respectively.

To better understand CACS we first consider two extreme design points. At one end of the spectrum, we have *uniform random sampling*, i.e. $\pi_i = \frac{m}{n}$ when m nodes are to be sampled. Uniform random sampling is agnostic to cost, as in the existing CS literature. At the other extreme is *greedy sampling* that selects one sample after another, always picking the sample with lowest costs, i.e. $\pi_i = 1$ if c_i is among m smallest and $\pi_i = 0$ otherwise.

The following examples illustrates the limitations of these two extremes and motivate the need for intelligent CACS.

We compare two scenarios: 1) costs are i.i.d among samples, shown in Fig. 1(a); and 2) sample costs are spatially correlated (typical case in sensor networks and crowdsensing), shown in Fig. 1(b). In the figures, the left side illustrates the sampling cost map, with lighter color indicating larger cost; and the two scenarios have the same overall cost distribution. The right side plots the cost-accuracy tradeoff of greedy and random sampling, respectively. We observe that when the costs

are i.i.d., the greedy algorithm performs very well because it anyway selects a set of samples with sufficient randomness. At 80% accuracy, its cost is only 20% of random sampling, and only 5% of the total cost. However, when spatial correlation exists, the performance of greedy significantly deteriorates; the selected samples no longer allow an accurate recovery. In fact, for the same accuracy, greedy sampling is actually more expensive than random sampling (20% more at 80% accuracy).

These examples show that it is non-trivial to design a sampling strategy that is inexpensive *and* guarantees accurate recovery in all cases. The greedy strategy favors low-cost samples, and performs poorly in cases when the cost map contains spatial correlation; random sampling only considers recovery accuracy and disregards cost factors. Therefore, the first challenge of CACS is to find the best possible balance between these two extremes. Its design and implementation is the focus of this paper.

It is worth noting that another potential approach is to construct an RIP-satisfying sensing matrix via a deterministic sampling strategy [9, 1, 4, 17], e.g. via sophisticated coding techniques. While such an approach is beyond the scope of this paper, we believe that such strategies are hard to combine with sampling costs because it seems that either RIP would be violated, or – for certain worst-case cost-maps – deterministic sampling strategies can be extremely expensive.

3.3 Challenge: Predicting Recovery Accuracy

In order to balance recovery accuracy and sampling cost, we need to quantitatively analyze these two factors. For a given cost map, the benefit in cost saving of a sampling strategy is easy to calculate. However, especially for compressible signals, quantifying the recovery accuracy is often a posteriori, and one typically needs the ground truth to determine the exact accuracy.

RIP, as introduced in §2, is widely used to prove theoretical guarantees on the recovery error for given recovery algorithms (see [13]).

However, RIP has its limitations and is not a practical mechanism to predict recovery accuracy in our case. First, verifying RIP of a matrix is computationally inefficient and has been proven NP-hard [2]. Secondly, the existing approach to construct partial Fourier matrix that satisfies RIP only provides an insufficiently loose lower bound on the recovery accuracy. For example, it is shown that $\sim k \log^4 n$ samples are required for accurate recovery, while only $k \log n$ typically suffice in most practical scenarios [5].

Thus, a key challenge of CACS derives from the difficulty to *predict* the recovery accuracy of a given sensing matrix, let alone in an efficiently way. As a result, lowering cost and simultaneously keeping recovery accuracy becomes a harder. This challenge influences our design of CACS.

4. ALGORITHMIC FOUNDATIONS

In the following section, we detail the algorithmic underpinnings of CACS framework.

4.1 From RIP to Regularized Column Sum

Motivated by these challenges, we first have to find feasible metric of a sensing matrix that can be use as an indicator for recovery accuracy. Instead of the sophisticated RIP conditions, [5] proposed the Statistical RIP (StRIP and UStRIP condition), a probabilistic version of RIP, which

is much easier to verify and can still provide sufficiently good performance guarantees in most practical scenarios. In [5], three simple conditions are formulated to verify StRIP, i.e. 1) orthogonality and zero-sum rows, 2) “pairwise multiplication” condition, and the most important 3) bounded column sum (see [5] for details). For Fourier ensembles, Condition 1 and 2 are naturally satisfied, and the *bounded column sum* plays an essential role. We re-formulated the condition below under the term *Regularized Column Sum* (RCS).

DEFINITION 1. (*Regularized Column Sum*)
Let $\Omega \subseteq \{1, \dots, n\}$ denote the index set of a sample instance, and $|\Omega| = m$. Let $\mathbf{F}_\Omega = \{f_\Omega(j)\}$ denote the partial Fourier matrix containing the rows indexed by Ω . The *Regularized Column Sum* (RCS) of \mathbf{F}_Ω is

$$\sigma(\mathbf{F}_\Omega) = \max_{j=2, \dots, n} \log_m \left| \sum_{i \in \Omega} f_{i,j} \right|^2, \quad (2)$$

where $f_{i,j}$ is the element in i^{th} column, j^{th} row of \mathbf{F}_Ω .

The next lemma explicitly recalls the main results of StRIP and shows how the RCS and RIP conditions are connected.

LEMMA 1. (*Statistical RIP*)
For a sensing matrix \mathbf{F}_Ω with $\eta \in (0.5, 1]$, which satisfies

$$\eta \leq 2 - \sigma(\mathbf{F}_\Omega), \quad (3)$$

there exists a constant $c > 0$ such that, if $m \geq (c \frac{k \log n}{\delta^2})^{1/\eta}$, with probability $1 - \epsilon$, $\frac{1}{\sqrt{m}} \mathbf{F}_\Omega$ satisfies RIP of order $\forall k < 1 + (n - 1)\delta$, with isometry constant δ , where $\epsilon = 2 \exp \left[-\frac{[\delta - (k-1)/(n-1)]^2 m^\eta}{8k} \right]$. The probability is with respect to all k -sparse vectors uniformly drawn from the space.

In Lemma 1, a smaller ϵ indicates a higher probability of satisfying RIP. When the sparsity of the signal k is fixed, a larger η implies a smaller ϵ . That requires a smaller $\sigma(\mathbf{F}_\Omega)$. That is to say, a sensing matrix with a smaller RCS has better chance to satisfy RIP.

By integrating Lemma 1 and Theorem 1 in [7], we have the following theorem that bridges the recovery accuracy bound and RCS when using ℓ_1 -min as the recovery algorithm.

THEOREM 1. (*RCS-Recovery Bound*)
A compressible signal in Fourier domain $\mathbf{y} = \mathbf{F}\mathbf{x}$, has a bounded perturbation from a k -term signal $\mathbf{y}_k = \mathbf{F}\mathbf{x}_k$, i.e. $\|\mathbf{y} - \mathbf{y}_k\|_{\ell_2} \leq e$. For a partial Fourier matrix \mathbf{F}_Ω , $|\Omega| = m$, let $\hat{\mathbf{x}} = \arg \min_{\hat{\mathbf{x}} \in \mathbb{R}^n} \|\hat{\mathbf{x}}\|_{\ell_1}$, s.t. $\|\mathbf{F}_\Omega \hat{\mathbf{x}} - \mathbf{y}_\Omega\|_{\ell_2} \leq e$. Then $\hat{\mathbf{x}}$ satisfies

$$\|\hat{\mathbf{x}} - \mathbf{x}_k\|_{\ell_2} \leq \frac{C_k e}{\sqrt{m}}$$

with probability at least 0.99, there exists $c > 0$, $\forall k < \frac{n}{8}$, if $\sigma(\mathbf{F}_\Omega) < \min\{\frac{3}{2}, A_1, A_2\}$, $A_1 = 2 - \log_m(4ck \log n)$, $A_2 = 2 - \log_m \left[678.19k \left(\frac{n-1}{n-8k+1} \right)^2 \right]$, the constant C_k only depends on the RIP constant.

Theorem 1 states that for a given sparsity k , if $\sigma(\mathbf{F}_\Omega)$ is small enough, with an overwhelming probability, the recovery error in ℓ_2 -norm is upper bounded. The theorem establishes RCS as an indicator for recovery accuracy. Most importantly,

RCS of a matrix is easy to calculate, and Theorem 1 can be verified in $O(mn)$ time. The proof of the theorem can be found in the Appendix, and the complete proof is presented in the full version of this paper [31].

The recovery accuracy bound in Theorem 1 is given for ℓ_1 -min recovery algorithm, and similar results can also be established for other recovery algorithms by setting proper parameters.

4.2 CACS via Convex Optimization

According to Theorem 1, given a certain level of recovery accuracy requirement, one can always find a constant α such that when $\sigma(\mathbf{F}_\Omega) \leq \alpha$, the required recovery accuracy is met with high probability. Given this intuition, the next step is to find a satisfying sampling matrix with the lowest cost.

However, for a given α , the computational complexity of finding an *exact* sampling matrix Ω with minimal total cost that satisfies the constraint $\sigma(\mathbf{F}_\Omega) \leq \alpha$ is exponential. To address this challenge, we recall that CACS actually conducts a randomized sampling process and thus we modify the constraints to $\mathbb{E}[\sigma(\mathbf{F}_\Omega)] \leq \alpha$. Accordingly, this modification leads to a probabilistic version of Theorem 1 via Markov Inequality. This is equivalent to relaxing a $(0, 1)$ -integer program (finding $\pi \in \{0, 1\}^n$) into a linear program (finding $\pi \in [0, 1]^n$). It is worth mentioning that plenty of works have studied the gap between integer programming and its linear relaxation [24]. However, our focus is the *average* performance instead of worst case performance.

In summary, we formulate the following convex optimization problem, called *RCS-constrained Optimization* (RO), to find an optimal randomized sampling strategy π which satisfies the given RCS constraint with the lowest cost.

$$\begin{aligned} \text{(P)} \quad & \underset{\pi}{\text{minimize}} && \mathbf{c}^T \pi \\ & \text{subject to} && \mathbf{1}^T \pi = m \\ & && (\text{Re}(\mathbf{F}_{\cdot j})^T \pi)^2 + (\text{Im}(\mathbf{F}_{\cdot j})^T \pi)^2 \leq \alpha^2 \\ & && 0 \leq \pi_i \leq 1, \quad i = 1, \dots, n. \end{aligned}$$

where \mathbf{c} is the cost map, π is the sampling strategy, m is the expected sample size, and $\text{Re}(\mathbf{F}_{\cdot j})$ and $\text{Im}(\mathbf{F}_{\cdot j})$ denote the real and imaginary component of the j^{th} column in \mathbf{F} respectively.

Note that lowering the sampling cost and RCS are conflicting objectives. Therefore RO introduces a new constraint on the expected sample size m , and employs it as the proxy to balance RCS and the sampling cost. The output of RO is a randomized sampling strategy π which also means only the *expected* sampling cost is minimized while the *expected* RCS is constrained. Although this is not as strong as a deterministic guarantee, the recovery bound is still met with high probability and the expected sampling cost reflects the average performance of a sampling strategy, as we discussed earlier. Attractively, RO can be solved in polynomial time via standard interior point methods [16].

4.3 Performance Guarantee

The parameter α , representing the value of RCS, plays a critical role in the performance of the sampling matrix: 1) the bigger the value of α , the more greedily the algorithm

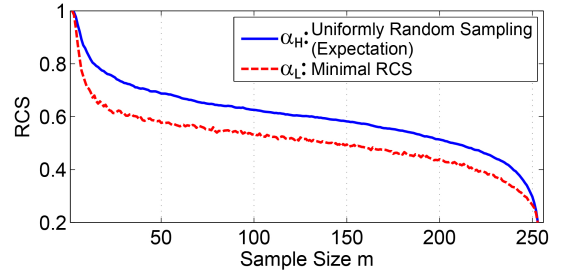


Figure 2: Expected RCS of Ransom Sampling and Minimal RCS

behaves, 2) the smaller the value of α , the better the recovery accuracy. Therefore, by adjusting the value of α , we can control the behavior of RCS-constrained Optimization. Thus a good choice for α should ensure the following points:

- There exists solutions to (P), i.e. α should be large enough so that the problem is feasible.
- The recovery guarantee is no worse than uniformly random sampling, i.e. α should not be too large or we lose too much accuracy guarantee.

Figure 2 illustrates the changes in $\mathbb{E}[\sigma(\mathbf{F}_\Omega)]$ of uniformly random sampling (denoted by α_H), and the minimal – found via Simulated Annealing – RCS (denoted by α_L) as functions of the sample size m with $n = 256$. Both curves monotonically decrease regardless of the value of n . Therefore, one can use m as a proxy to balance the tradeoff between recovery accuracy and sampling cost. Furthermore, if we choose $\alpha \in [\alpha_L, \alpha_H]$, it satisfies the above mentioned two conditions.

THEOREM 2. (RO Performance Guarantee)

Given a sample size m and $\alpha \in [\alpha_L, \alpha_H]$, RCS-constrained Optimization outputs a randomized sampling strategy π that satisfies:

- The recovery error bound of π is no worse than that of the uniform random strategy given the same sample size.
- The expected cost of π is the lowest among all randomized sampling strategies that satisfies the RCS constraint.

PROOF SKETCH. We need to show that, 1) the expected RCS of output π is smaller than that of uniform random sampling, thus making π achieving a better or equal recovery accuracy bound according to Theorem 1; and 2) we can always find a solution to (P) provided $\alpha \geq \alpha_L$, because the solution space of the convex optimization is not empty. Moreover the solution is with the least cost. \square

4.4 Tuning the Parameters

Theorem 2 provides performance guarantees of RO, but does not quantify its cost-benefit or determine how α should be set to obtain the solution. In this section, we illustrate the impact of α on different types of cost maps and then discuss how to choose a good value of α in practice.

Fig. 3 shows the cost-accuracy tradeoff of RO on three cost maps when changing the values of α . The arrow shows the change from α_L to α_H on each curve. We can observe that RO behaves similarly to random sampling when $\alpha = \alpha_L$, and RO performs like greedy sampling when α grows to α_H . On i.i.d cost map, it is clear that greedy sampling is the best since it spends least cost while achieving a best recovery accuracy, and RO performs better as α grows. On

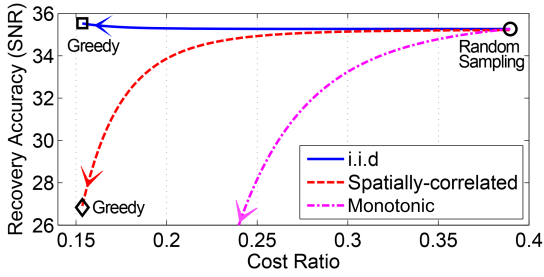


Figure 3: Impact of α

the spatially-correlated cost map, the sampling cost decreases while the recovery accuracy almost remains at the same level as α grows initially; and then beyond a certain point, the accuracy drops quickly. Thus on spatially correlated cost maps, RO is able to always find a sweet-spot between cost and accuracy provided an appropriate value of α . On the monotonic cost map (the cost increases monotonically from one side of the map to the other), the recovery accuracy rapidly decreases as α grows, which means it is difficult to find a good balance between cost-efficiency and recovery accuracy. This figure illustrates that the cost-accuracy tradeoff depends on the spatial correlation of the cost map. Furthermore, this tradeoff can be used as a good indication of average performance.

In practice, natural signals often exhibit periodic correlation (e.g., traffic conditions), which allows us to utilize historical data for tuning parameters. In particular, a scatter diagram containing cost-accuracy tradeoffs of different values of α can be drawn and one can select an appropriate value of α to achieve the desired balance of accuracy and sampling cost.

4.5 Distributed Heuristics

Although RO is attractive in many aspects, it is essentially a centralized algorithm, and thus has practical limitations. It requires the *global information* of the cost map to perform its optimization; it may not be able to adapt fast to dynamic cost maps; and it is difficult to be embedded into mobile crowdsensing systems without a central controller.

For this reason, we design two decentralized, low complexity sampling algorithms for distributed scenarios: *Distributed Weighted Sampling* (DWS) and *Pairwise Sampling* (PW). They are designed in a distributed fashion, and only require partial information, local communication, and lightweight computation. We also analyze their potential cost benefit and their RCS performance.

4.5.1 Distributed Weighted Sampling

Inspired by the pioneering results in [27], we suggest a more general *Distributed Weighted Sampling* (DWS) scheme where the weights are set to be inversely related to the corresponding sampling cost at different nodes. In this way, we expect the final samples to preserve a certain degree of randomness while samples with lower costs are nevertheless favored. In particular, the probability of choosing a node is

$$\pi_i \propto \frac{c_i^{-\beta}}{\sum_i c_i^{-\beta}}, \quad \sum_i \pi_i = m, \quad (4)$$

where m is the expected sample size, and the weighing parameter β balances accuracy and cost. Notice that when $\beta = 0$, DWS becomes uniform random sampling, while when

Algorithm 1 Distributed Weighted Sampling

INPUT : Weighting parameters β , sampling threshold Θ , and cost map $\{c_i\}$
OUTPUT: Sample index set Ω
for node i **do**
 $r_i = \text{random}(0, 1)$; $k_i = r_i^{1/c_i^{-\beta}}$;
 if $k_i \geq \Theta$ **then**
 $\Omega \leftarrow \Omega \cup \{i\}$;
 end if
end for
Return Ω .

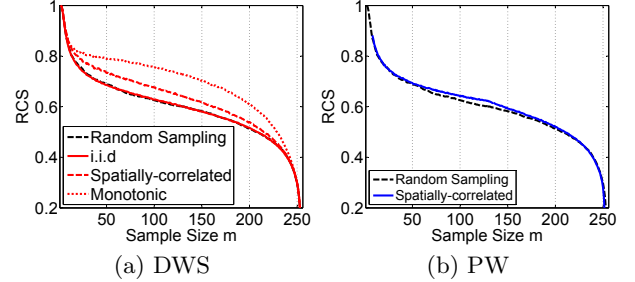


Figure 4: RCS of the Distributed Algorithms

$\beta \rightarrow \infty$, it turns into greedy sampling. Therefore, we can use β to balance cost and accuracy performance based on the cost distribution.

The design of DWS is presented in Algorithm 1. The distributed implementation of DWS follows the philosophy of [12], in which an algorithm of weighted random sampling without replacement is proposed. Generally speaking, each node i first generates a random number $r_i \sim U[0, 1]$, and calculates the opportunity value $k_i = r_i^{1/c_i^{-\beta}}$. In order to obtain m samples according to the underlying weights $\{c_i^{-\beta} / \sum c_i^{-\beta}\}$, nodes with the m largest k_i should be sampled. Instead of collecting k_i from participating nodes, which leads to heavy communication overhead, a *sampling threshold* Θ is distributed to all nodes, and nodes that have $k_i \geq \Theta$ report themselves as samples. In fact, Θ is an empirical value of the m^{th} largest opportunity value learned from historical data.

RCS: Fig. 4(a) shows the value of RCS as a function of m for DWS ($\beta = 1$) in the three different cost maps. We also plot the RCS of uniform random sampling as a baseline. We note that in the figure, under the i.i.d. cost map, RCS of DWS is almost the same as that of random sampling; i.e., the preserved randomness among samples helps DWS keep a comparable recovery guarantee to random sampling. In the case of the spatially-correlated map, the gap is slightly larger (while we expect lower cost of DWS as analyzed next). In the case of the monotonic cost map which exhibits extreme spatial correlation, it is natural to observe RCS is much higher than random sampling because DWS favors lower cost samples.

Cost Benefit: Assume that the cost map is i.i.d. with density function f_c . The expected cost of random sampling is $\mathbb{E}_R(c) = \int_0^\infty x f_c(x) dx$, and thus the expected cost for a sample is

$$\mathbb{E}_{\text{DWS}}(c) = \int_0^\infty \frac{x^{1-\beta} f_c(x)}{\int_0^\infty y^{-\beta} f_c(y) dy} dx.$$

Table 1 summarizes the cost saving in percentage for two representative cost distributions: uniform and exponential

Table 1: Cost Saving of DWS

β	0	0.1	0.5	1	5
U[1,100]	0%	5%	27%	57%	97%
Exp[1]	0%	5%	48%	84%	99.9%

distribution. The values provide intuition on the degree of cost benefits in different scenarios. The savings are more significant in the exponential case, which is due to the fact that the exponential distribution has more samples with low cost and thus provides more room for improvement.

4.5.2 Pairwise Sampling

DWS is a simple and intuitive sampling algorithm. However, there are situations where its performance may be far from optimal. Specifically, observe that DWS favors low-cost samples. Therefore, when costs are strongly correlated, e.g., monotonically increasing in one direction spatially, the cost saving benefits of DWS diminishes, because the randomness in the selected samples is largely lost (see Fig. 7(a) in §6 as an example). To address this problem, we propose an alternative sampling algorithm, *Pairwise Sampling* (PW) that prioritizes recovery accuracy over lower cost.

PW generally follows 3 steps in its sampling process:

1) Pick $2m$ sample candidates uniformly at random; 2) Select m pairs of candidates such that the candidates in each pair are within each other’s local communication range r ; 3) Choose the candidate with lower cost in each pair as a sample. The recovery accuracy is ensured by step 1 and 2, since pairs are made according to local communication, uniform randomness is preserved, thus leading to samples that resemble a uniform random strategy. The cost-saving benefit is achieved in step 3 – the power of two choices.

However, in a distributed system, it is hard to form exactly m pairs with only local communication. Therefore, we slightly modify the original PW, as shown in Algorithm 2. For each candidate, it will search for another candidate within the local communication range. Pairs are formed when a candidate first finds another unpaired candidate within its local communication range. If there is no other unpaired candidate in its local communication range, take this candidate as sample with probability 50%.

RCS: Fig. 4(b) shows the value of RCS via PW in the three cost maps compared with random sampling. We see that the RCS of PW and random sampling are almost identical, which is highly desirable. The RCS performance of PW is consistent in different cost distributions. To better distinguish the RCS curve of PW from random sampling, we omitted RCS curves of PW in other distributions.

Cost Saving: We can study the sampling cost of PW similarly as in the case of DWS. Supposing the costs are i.i.d. and obey a certain distribution with density function f_c , the expected cost for one PW sample is

$$\mathbb{E}_{\text{PW}}(X) = 2 \int_0^\infty x f_c(x) \int_x^h f_c(y) dy dx.$$

Similarly, we find that the cost saving of PW is 33.3% in uniform distribution $U[0,1]$, and 50% in exponential distribution $Exp[1]$ compared to random sampling.

4.5.3 Comparison between DWS and PW

Uniform random sampling and greedy sampling are two extreme ways of dealing with sampling costs. DWS and PW balance the two extremes from different angles. DWS is

Algorithm 2 Pairwise Sampling

INPUT : Cost map $\{c_i\}$, and local transmission range r
OUTPUT: Sample index set Ω
Choose $2m$ seeds $\mathcal{W}_s \subset [n]$ uniformly at random.
for all $i \in \mathcal{W}_s$ **do**
 $\mathcal{R}_i = \{j \in \mathcal{W}_s, \text{distance}(i, j) \leq r\}$
 if $\mathcal{R}_i = \emptyset$ **then**
 $\Omega \rightarrow \Omega \cup \{i\}$ with probability 0.5.
 $\mathcal{W}_s \rightarrow \mathcal{W}_s - \{i\}$
 else
 randomly pick $j \in \mathcal{R}_i$
 if $C_i \leq C_j$ **then**
 $\Omega \rightarrow \Omega \cup \{i\}$
 else
 $\Omega \rightarrow \Omega \cup \{j\}$
 end if
 $\mathcal{W}_s \rightarrow \mathcal{W}_s - \{i, j\}$
 end if
end for
return Ω

a greedy-like method in which we give higher priority to candidates with lower costs, i.e., the greediness is adjusted by the parameter β . On the other hand, PW prioritizes achieving lower RCS over cost saving, and thus, with the same number of samples, it achieves a recovery accuracy comparable to random uniform sampling. In summary, the advantages of DWS are i) a naturally decentralized implementation without local communication and ii) its cost saving is significant when the cost distribution is not highly correlated with the underlying data. The advantages of PW are i) for a desired recovery accuracy, it is easy to compute the number of samples needed and ii) it performs better when costs have high spatial or temporal correlation.

5. SYSTEM IMPLEMENTATION

In this section, we conclude the description of CACS by detailing the design of its prototype implementation.

Fig. 5 illustrates the current CACS implementation. It consists of a mobile component, currently running on standard Android devices, and a web component using .Net-based web service.

5.1 Mobile Component

On the mobile side, sampling is triggered by sensor probe and implemented by “Compressive Sensing Sampler” using cost estimate obtained from “Mobile Cost Estimator”; and then delivered by the network interface. In addition, a participant uses the “User Configuration” module to flexibly configure other modules to meet the needs under different scenarios. The major modules are detailed as follows:

Sensor Probe. Operating as an event-driven Android service, Sensor Probe wakes based on OS-level system events that fire each time an application or daemon is initiated. A probability threshold (ϕ) is used to regulate how often Sensor Probe proceeds to waking Compressive Sensing Sampler, which would then decide if a sensor sample is to be collected.

Compressive Sensing Sampler. A stream of sensor sampling opportunities (as regulated by Sensor Probe) are evaluated based on their associated resource costs and their contribution to the randomness of the pool of previously collected samples. As already described, DWS is an approach based on weighing sampling opportunities using a candidate weight function (4).

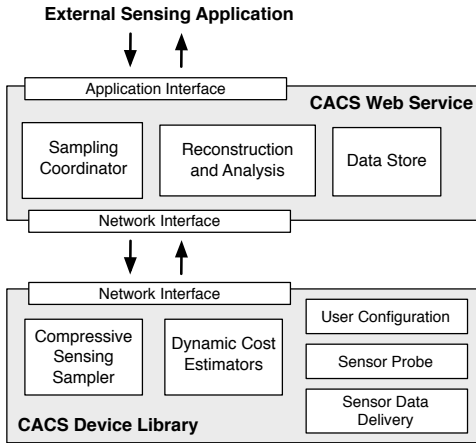


Figure 5: CACS Implementation

Mobile Cost Estimators. We introduce a cost estimation function into the process of CACS, which estimates the value of sampling cost in absence of the real cost values. Our current implementation supports the estimation of various cost types (viz. GPS sensor, cellular transmission, battery usage/perception).

Estimation of each cost type is performed by a separate estimation module that relies on the same overall design. Specifically, multi-factor regression models are trained for each module to estimate the current cost of an operation based on the *recent* energy costs of performing this same operation[3]. The influence of prior cost observations is discounted using an exponential decay function to allow for the decrease of relevance as time elapses. This design attempts to target the spatial and temporal relationship in the cost of operations such as sampling sensors or transmitting data. In addition to historical data, we use data like coarse location (provided at low-energy cost from cell towers) and categorical variables such as the type of wireless network being used (e.g., 3G or WiFi) to further improve estimation accuracy. Although our current estimator design is fairly simple, we find in experiments they have acceptable level of accuracy (see §6.3).

Sensor Data Delivery. Two modes of data delivery are supported by CACS. First, a near real-time delivery mode in which sensor data is transferred using either a cellular or WiFi network interface soon after sampling. Under this delivery mode the cost of transmission is already considered by Compressive Sensing Sampler when the sampling decision is evaluated. Second, a delay tolerant delivery mode is implemented in which collected data is stored locally and delivered when resource costs are minimized; specifically, a heuristic is used that transfers when WiFi is available and the phone is line-powered.

5.2 Web Component

When sampling data is delivered to the web service, it is stored at the data storage module. The data is then processed by the "recons" to reconstruct the phenomena of interests. It is also passed to the sampling coordinator module to further guide data sampling on mobile devices.

Reconstruction and Analysis. As mentioned in §4.1, recovery accuracy is guaranteed when using ℓ_1 -min. However, in practice, we instead employ Orthogonal Matching Pursuit

(OMP) to perform the reconstruction process because it is faster (OMP - $O(kmn)$ vs. ℓ_1 -min - $O(n^3)$). Furthermore, we notice better practical performance of OMP compared to ℓ_1 -min despite the lack of theoretical guarantees of OMP. The reconstructed data is stored for future analysis, e.g. parameter tuning.

Sampling Coordinator. Sampling Coordinator is executed upon the arrival of any data from participating mobile devices. The parameter values (such as Θ and β in DWS and r in PW) that need updates are propagated to mobile devices within the ACK payload sent in response to uploaded sensor data. As a result, mobile devices are updated with negligible overhead; the only negative consequence being that devices may operate with slightly different values of these parameters for short periods of time.

6. EVALUATION

We evaluate the performance of CACS algorithms using real-life data traces and experiments.

6.1 Methodology and Datasets

Two concrete application scenarios are considered to evaluate the important aspects of CACS and our algorithms thoroughly, including the performance, impact of different cost maps, robustness, and impact of various parameters.

6.1.1 Datasets

We first consider an *air pollution monitoring system* in Beijing that contains statically deployed sensors and a central server which collects data readings from sensor nodes and directs the behavior of them. The dataset of the *Air Quality Index* (AQI) contains hourly snapshots of the air quality map in the city covering an urban area of 32×28 km over several months in 2013. In each snapshot, 256 PM2.5 AQI readings aligned in a 16×16 equally spaced grid are used as ground truth. Sensors report their data readings directly to the central server via the 3G cellular network.

In this scenario, we use the 3G network transmission energy consumption as cost. The cost map is shown in Fig. 6(a). The central server decides the choice of sensors based on the cost map and CACS, and the chosen sensors report their data readings. After receiving the samples, the central server recovers the air pollution map for the entire area. (Other cost maps are also evaluated using synthetic data.)

Two weeks' data is used for evaluation. Each algorithm trains its parameters on the first week, and evaluates the performance based on the second week's data. The average result of multiple rounds is presented.

The second scenario is *traffic monitoring*, a representative mobile crowdsensing application. The dataset contains traffic information over Beijing city for three weeks in May, 2009 [28]. There are more than 8500 moving devices (taxis) deployed during the experiment period, where GPS and speed are recorded every 60 secs. There are over 20 millions records in each week.

Main roads in the urban areas are divided into small segments with a length of 200m. The 30-min average speed on each road segment during typical work hours in weekdays (8am-9pm/Mon-Fri) are used as the ground truth. The impact of different settings in this scenario are discussed in [33].

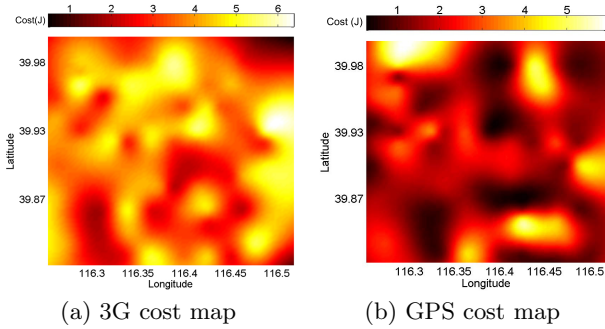


Figure 6: 3G/GPS cost map

6.1.2 Cost Functions

Here we conduct outdoor measurements of GPS energy consumption using a Google Nexus One devices at hundreds of evenly distributed places across Beijing. Meanwhile we also record the signal strength of 3G cellular network at those locations, and approximated the corresponding energy consumption based on the model in [10]. Fig. 6 illustrates the 3G and GPS energy consumption map generated from our experiments.

Furthermore, we use the battery traces of over 3000 mobile phones over several months as the temporal cost data. The remaining battery level of a device can be naturally considered as a type of “*perception cost*” – the lower the remaining battery, the more valuable it is, the higher cost it should be assigned - i.e., a crowd based system should be more cautious to sample devices with a low battery level.

We define $Cost(b) = B^{1-b}$ as the perception-based cost function for the remaining battery, where b is the ratio of the remaining battery, and B is a constant. In particular, as b goes to zero, the cost is high and approaches B quickly. We choose B to be comparable to the maximum GPS sampling cost ($B = 10$ in our evaluation). The intuition is that when b is large, users are not sensitive and thus other costs dominate. On the other hand, when b is small, users are sensitive and thus this factor dominates.

In many wireless sensor/mobile crowdsensing scenarios, both location information and battery status are essential. On the other hand, transmission latency (from minutes to days) may be tolerable, depending on the specific application scenario. Therefore, we choose a linear combination of spatial cost maps and the battery penalty as the overall sensing cost.

6.1.3 Performance Metrics

We use recovery accuracy and sampling cost as performance metrics to evaluate our proposed algorithms. First, we employ the *Normalized Mean Squared Error* (NMSE) in ℓ_2 -norm as the accuracy metric. Specifically, supposing $\hat{\mathbf{y}}$ is the recovery of data \mathbf{y} , the NMSE is defined as:

$$NMSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|_{\ell_2}^2}{\|\mathbf{y}\|_{\ell_2}^2}.$$

Furthermore, we use the *cost ratio* to measure sample cost, which calculates the proportion of chosen samples’ cost over the sum of cost of all candidates:

$$\text{Cost Ratio} = \frac{\sum_{\text{chosen } i} c_i}{\sum_{\text{all } i} c_i}.$$

6.1.4 Baseline Algorithms

Four alternative sampling strategies are also implemented and evaluated as comparison baselines:

- **Random:** samples are uniformly randomly chosen, i.e. $\pi_i = \frac{m}{n}$ where m is the expected sample size.
- **Greedy:** samples with lowest cost are chosen, i.e. $\pi_i = 1$ where c_i is among the m lowest cost, and $\pi_i = 0$ otherwise.
- **Non-uniform CS (NCS):** a state-of-the-art non-uniformly random sampling strategy [27].
- **CS-UTS:** only for traffic monitoring scenario, a state-of-the-art CS-based approach for traffic monitoring [33]. A subset of the *devices* are chosen, and all records from them are collected.

6.2 Overall Performance

The performance comparison is shown in Fig. 7, where the cost ratio of achieving three different accuracy levels (NMSE=10%/5%/2.5%, stands for adequate/high/extreme accuracy) are presented. The 3G and GPS cost maps and their combination are tested for AQI data, while for traffic data, cost maps with or without battery penalty are evaluated. The values of β in DWS are set to be optimal in each scenario via the method discussed in §6.4.1.

As we can observe from the figure, RO outperforms all baseline algorithms in all scenarios, e.g., in Fig. 7(b) at NMSE=5%, RO is 63.0%, 52.4% and 54.5% lower compared with random, greedy and NCS respectively. Generally, RO saves 30% ~ 70% cost in adequate/high accuracy levels, while the cost benefit shrinks at extreme accuracy levels because it requires a large number of samples to achieve this level. For instance, in the traffic scenario, more than 60% of the total cost is consumed and, in fact, more than 70% of nodes are sampled. However in such cases, RO still outperforms all baselines.

The two distributed algorithms also present considerable cost savings under all scenarios. For example, in Fig. 7(e) at NMSE=10%, DWS saves 35.7%, 57.1%, 25.0%, and 47.1% cost compared with random, greedy, NCS, and CS-UTS baselines. Similarly, in Fig. 7(a) at NMSE=2.5% PW costs 20.0%, 41.7%, and 17.6% lower than random, greedy and NCS baselines respectively. Occasionally PW outperforms RO because it can provide better recovery accuracy. Next, we investigate the impact of different cost factors separately.

6.2.1 The Impact of Spatial Correlation

As discussed earlier, the performance of those algorithms that make decisions highly based on cost, namely DWS and greedy sampling, may deteriorate when cost map has high spatial correlation, which leads to poorly conditioned sensing matrices thus fail to accurately recover the original signal. This is clearly shown in Fig. 7(a)~7(d) where greedy sampling performs poorly and DWS barely outperforms random sampling, meanwhile the NCS baseline also suffers from the same problem.

On the contrary, RO and PW have a much better ability to balance cost and accuracy in such cases. Thus they also exhibit much better performance, especially at high accuracy levels. It is worth mentioning that even in such cases DWS can still outperform all baselines given properly tuned parameters based on historical data.

Both the GPS and 3G energy cost map show clear spatial correlation (Fig. 6). Further in those cases when only spatial

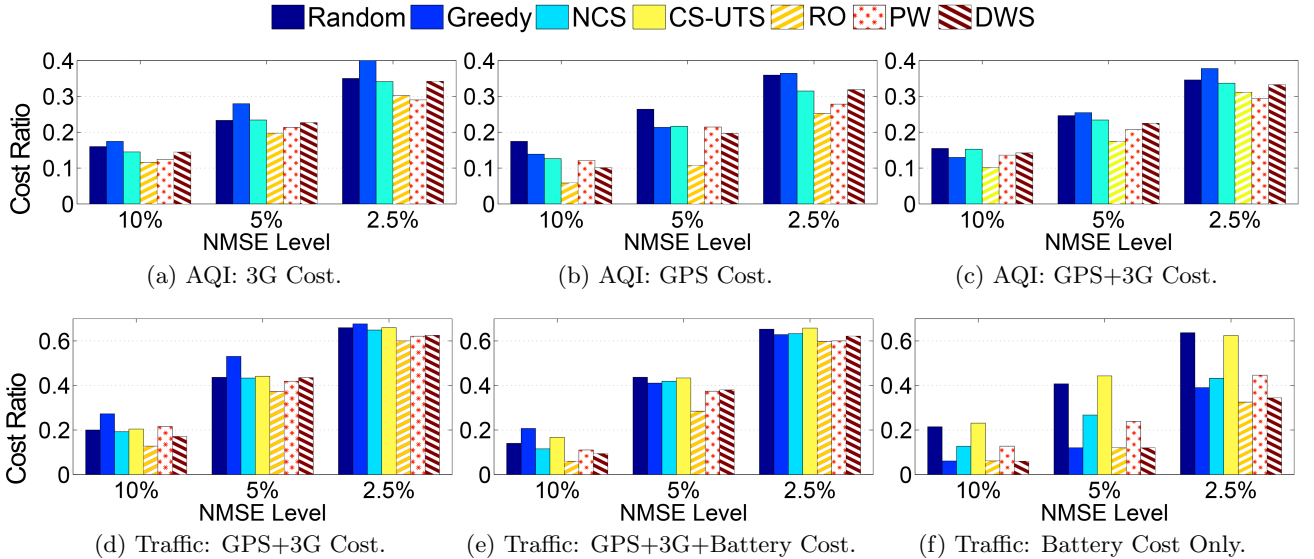


Figure 7: Performance Comparison

cost is considered, we can find DWS has better performance when using the GPS cost map ($\sim 10\%$ improvement) because the 3G cost map has stronger spatial correlation. This is also supported by the fact that greedy sampling outperforms random sampling occasionally in Fig. 7(b) at NMSE=10% and 5%.

6.2.2 The Impact of the Battery Perception Cost

Different from the highly correlated spatial cost map, the battery penalty cost in our scenario introduces a level of independency among devices. In such scenarios, high cost-dependency in battery traces helps greedy-like algorithms in reaping the benefit of cost-awareness, and this explains why in Fig. 7(e)~7(f) greedy and DWS perform much better than in the other cases. On the other hand, RO and PW can also continuously provide stable high accurate recovery with lower cost in these mixed-battery cases.

Especially in Fig. 7(f) where only battery cost is considered, by taking the advantage of the inherent randomness within the cost map, DWS, NCS, and greedy sampling outperform CS-UTS, PW and random sampling at all accuracy levels. Among these three greedy-like algorithms, DWS still surpasses the other two by 12%. In this case, PW falls behind due to limited saving on cost.

It is worth mentioning that we have evaluated three different cost functions, including log, linear, and exponential, and observe similar performance trends. We believe that the exponential one most accurately depicts users' preference, and thus omit other cases due to space limitations.

6.3 Robustness

In the previous experiments, all cost values are assumed to be accurate. However in practice, it is possible that error will be introduced when estimating or measuring cost values. These errors affect the performance of the algorithms that makes sampling decisions based on cost. Thus a good CACS algorithm should maintain robust performance when cost errors exist.

We introduce random noise of different levels (measured in ℓ_2 -norm) to various cost maps. RO, PW and DWS are tested

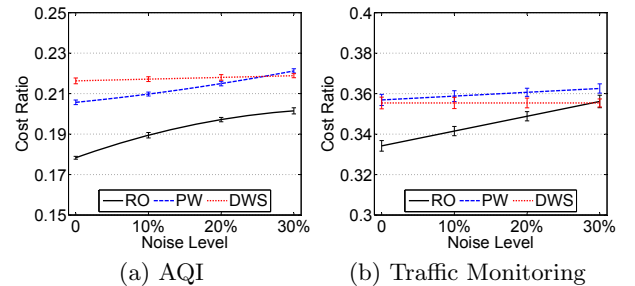


Figure 8: Robustness

over the noisy version of cost map while still using the true cost values to calculate the cost ratio of each algorithm. Table 2 shows the estimation error for GPS+3G cost map. Fig. 8 shows the changes in cost ratio at NMSE=5% as noise grows.

In the AQI scenario with GPS+3G energy cost map in Fig. 8(a), we can observe a slight increase in the cost ratio in three CACS algorithms, and the total increments are within 10%. In other words, all algorithms remain stable against the presence of noise in cost maps. Specifically, RO experiences a slightly rapid increase when noise is introduced but this rate later slows. PW has the largest growth in cost ratio, and surpasses DWS at the 25% noise level, while DWS almost keeps the cost ratio at the same level because the distribution of cost values is not changed significantly.

In Fig. 8(b), we observe significant resilience in terms of cost estimation errors in the proposed algorithms. In particular, with estimation error as high as 20%, which is higher than all estimation error in Table 2, we observe negligible difference.

6.4 Impact of Parameters

We focus on the parameters in DWS and PW in this evaluation in terms of both performance and practical impacts in distributed environments.

Table 2: Cost estimating error

Sampling rate	20%	30%	40%	50%
GPS+3G	16.2%	15.4%	14.6%	13.9%

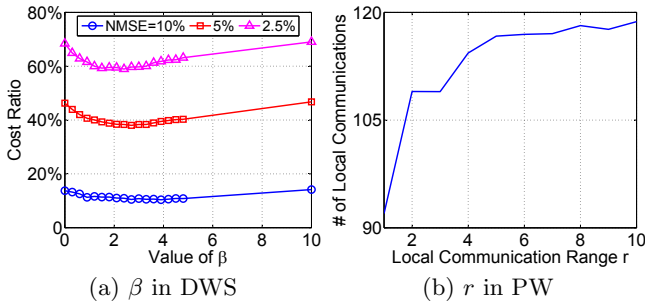


Figure 9: Impact of Parameters

6.4.1 Weighting Parameter β in DWS

As discussed in §4.5.1, the weighting parameter β alters the balance for DWS between randomness and greediness. Three curves in Fig. 9(a) present the change in cost ratio of DWS at NMSE=10%/5%/2.5% in the traffic scenario with GPS+3G+Battery costs, respectively. We can see a rapid decline in cost ratio, i.e. increase in performance, as β grows from 0. After a certain value, the cost ratio becomes larger as β increases. These curves demonstrate the impact of β on the performance of DWS, as well as the process of searching for the optimal choice of β . In addition, at different NMSE levels, the best β value also differs from each other, e.g. $\beta = 2$ is the best at NMSE=2.5% while $\beta = 3$ performs better at other levels.

6.4.2 Local Transmission Range r in PW

In Pairwise Sampling, the local transmission range r of each node in the network impacts the pairing procedure. This, in turn, has a direct impact on the overhead of local communication of PW as discussed in §4.5.2. In this evaluation, we present the change in local communication overhead with different r , where the overhead is measured as the number of local communications between nodes.

Fig. 9(b) illustrates the growth in local communications when sampling 60 nodes in the air pollution scenario under the GPS + 3G cost map applied. The local communications occur more and more frequently as the transmission range r grows. However, the increase slows down when r becomes larger. This illustration clear shows the tradeoff between PW efficiency and the local communication overhead.

7. DISCUSSION

We now discuss the generality of CACS, followed by the future research directions.

Generality. Although our evaluation is based on two specific scenarios, we anticipate CACS will also exhibit similar performance under a variety of other applications. Existing works have already shown that many modules commonly used in sensing systems (e.g., the accelerometer [32] and GPS [22]) have sparse representations. Furthermore, our cost framework can easily accommodate a variety of sampling costs other than those currently evaluated; e.g., costs associated with user attention and effort.

Future Directions. The proposed RCS-constrained Optimization (RO) algorithm has practical limitations – it assumes a static cost map with a central controller. Although we have proposed two distributed heuristics, neither of them has the theoretical performance guarantees. An improved

distributed version of RO is desirable as it would allow RO to apply to a wider range of scenarios.

More broadly, RCS-based analysis only works for partial Fourier matrices, while results of other types of bases are not yet studied. We believe that it is important to develop results for other bases widely used in CS-based applications, especially for use in scenarios where the Fourier base is not suitable. It also remains important to study the connection between RCS and *coherence*, another alternative metric whose relation with recovery accuracy has been studied [7, 1]. The challenge is that simple coherence is problematic to apply in most practical scenarios.

Finally, Theorem 1 provides the worst case performance guarantee. Analysis of the *average* performance is yet to be done. Such an analysis is non-trivial and is absent from the CS literature. However, progress in this direction is important as it is likely to directly lead to improvements in the resource efficiency of CACS.

8. RELATED WORK

Our study of cost-sensitive compressive sensing, as applied to networked sensing systems, touches upon a number of active areas of interest. In what follows, we describe the most salient related work in three key areas while also highlighting the novel contributions being made by CACS.

Theory of Compressive Sensing. Ever since its emergence [6, 11], a steady stream of compressive sensing applications continue to arrive. Research on CS can be divided into two categories. The first category focuses on the construction of the sensing matrix. Randomized construction is the mainstream [7, 5] approach. Several deterministic types of construction have also been proposed in recent years [9, 17]. The second category focuses on recovery algorithms such as BP and OMP [14, 19, 20]. CACS builds on these principled approaches but with a focus on the practical system challenges of the resource usage that is often lacking in this work.

Sensor Networks and Crowd Systems. Within the domain of network embedded systems, a strong body of work has explored the use of compressive sensing in static sensor networks, for example [18] and [29]. But as already explained this work has neglected a number of important dimensions related to resource costs when sampling.

Mobile crowdsensing is a relatively new emerging research area. Recently, a wide variety of application domains of mobile crowdsensing (e.g., [26, 23]) have been studied. Energy is an important issue, and significant effort has been invested in developing techniques to gather data in an energy-efficient manner, e.g., [21, 15, 22]. More closely related to CACS, traffic monitoring using CS has been explored in studies such as [33]. But in [33] resource cost are not factored into the process of sampling.

Compressive Sensing with Variable Sample Cost. A key contribution of CACS is the design – from the ground-up – of a CS framework that understands how costs can fluctuate when collecting different sensor data samples. Work is slowly building in CS that considers device costs, such as those encountered in sensor network deployments [8]. A more recent work has proposed methods for nonuniform sampling for use within sensing systems [27]. However, it does not quantitatively analyze the tradeoff between energy-saving and sensing accuracy. Similarly, [25] only assume costs are

fixed and known; as we have explained, CACS is designed to cope with both static and dynamic costs.

9. CONCLUSION

In this paper, we present CACS – a cost-aware compressive sensing framework for collecting large-scale sensor data from contributing devices, that consumes much lower amounts of resources (e.g., battery) than was previously possible. In particular, CACS recognizes *and is designed for* the wide fluctuations in resource costs that exist among sensor sampling opportunities. As a result CACS proposes principled algorithms for gathering sensor data with fewer high resource cost samples, yet critically maintains an adequate level of randomness to still support accurate data recovery under compressive sensing.

We evaluate CACS under two representative networked sensing scenarios – air quality and traffic monitoring – based on real-world large-scale datasets. Our findings show, for example, that CACS can achieve an 80% accuracy rate in data reconstruction, while only requiring 10% of the energy used by state-of-the-art compressive sensing approaches. Not only are our results significant for the particular application domains we study; but we also believe due to the strong algorithmic foundations of our approach, CACS is likely to generalize to a number of other domains of networked sensing.

Acknowledgement: We thank our shepherd Wen Hu and the anonymous reviewers for their insightful comments. This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003.

10. REFERENCES

- [1] A. Amini, V. Montazerhodjat, and F. Marvasti. Matrices with small coherence using-ary block codes. *Signal Processing, IEEE Transactions on*, 60(1):172–181, 2012.
- [2] R. G. Baraniuk. Compressive sensing. *IEEE signal processing magazine*, 24(4), 2007.
- [3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.
- [4] J. Bourgain, S. Dilworth, K. Ford, S. Konyagin, D. Kutzarova, et al. Explicit constructions of rip matrices and related problems. *Duke Mathematical Journal*, 159(1):145–185, 2011.
- [5] R. Calderbank, S. Howard, and S. Jafarpour. Construction of a large class of deterministic sensing matrices that satisfy a statistical isometry property. *Selected Topics in Signal Processing, IEEE Journal of*, 4(2):358–374, 2010.
- [6] E. J. Candès. Compressive sampling. In *Proceedings on the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, pages 1433–1452, 2006.
- [7] E. J. Candès, J. K. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.
- [8] C. T. Chou, R. Rana, and W. Hu. Energy efficient information collection in wireless sensor networks using adaptive compressive sensing. In LCN’09, pages 443–450, 2009.
- [9] R. A. DeVore. Deterministic constructions of compressed sensing matrices. *Journal of Complexity*, 23(4):918–925, 2007.
- [10] N. Ding, D. Wagner, et. al. Characterizing and modeling the impact of wireless signal strength on smartphone battery drain. In SIGMETRICS ’13, pages 29–40, 2013.
- [11] D. L. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.
- [12] P. Efraimidis and P. Spirakis. Weighted random sampling. In *Encyclopedia of Algorithms*, pages 1–99, 2008.
- [13] Y. C. Eldar and G. Kutyniok. *Compressed sensing: theory and applications*, 2012.
- [14] A. C. Gilbert, M. J. Strauss, et. al. One sketch for all: fast algorithms for compressed sensing. In STOC ’07, pages 237–246, 2007.
- [15] N. Lane, et al. Piggyback CrowdSensing (PCS): Energy Efficient Crowdsourcing of Mobile Sensor Data by Exploiting Smartphone App Opportunities. In SenSys ’13, 7:1–7:14, 2013.
- [16] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx>. 2014.
- [17] S. Li and G. Ge. Deterministic sensing matrices arising from near orthogonal systems. *Information Theory, IEEE Transactions on*, 60(4):2291–2302, 2014.
- [18] C. Luo, F. Wu, J. Sun, and C. W. Chen. Compressive data gathering for large-scale wireless sensor networks. In Mobicom ’09, pages 145–156, 2009.
- [19] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.
- [20] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of computational mathematics*, 9(3):317–334, 2009.
- [21] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In MobiSys ’10, pages 299–314, 2010.
- [22] P. Misra, W. Hu, et. al. Energy efficient gps acquisition with sparse-gps. In IPSN ’14, pages 155–166, 2014.
- [23] M.-R. Ra, B. Liu, T. F. La Porta, and R. Govindan. Medusa: A programming framework for crowd-sensing applications. In MobiSys ’12, pages 337–350, 2012.
- [24] P. Raghavan and C. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. *Combinatorica*, 7(4):365–374, Dec. 1987.
- [25] R. Rana, W. Hu, and C. T. Chou. Energy-aware sparse approximation technique (east) for rechargeable wireless sensor networks. In *Wireless Sensor Networks*, pages 306–321, 2010.
- [26] R. Rana, C. Tung Chou, et. al. Ear-phone: An end-to-end participatory urban noise mapping. In IPSN ’10, pages 105–116, 2010.
- [27] Y. Shen, W. Hu, R. Rana, and C. T. Chou. Nonuniform compressive sensing for heterogeneous wireless sensor networks. *IEEE Sensors Journal*, 13(6):2120–2128, 2013.
- [28] Traffic Data. <http://sensor.ee.tsinghua.edu.cn/download.php>. 2014.
- [29] X. Wu and M. Liu. In-situ soil moisture sensing: measurement scheduling and estimation using compressive sensing. In IPSN ’12, pages 1–12, 2012.
- [30] Y. Xiao, P. Simoons, P. Pillai, K. Ha, and M. Satyanarayanan. Lowering the barriers to large-scale mobile crowdsensing. In HotMobile ’13, pages 9:1–9:6, 2013.
- [31] L. Xu, X. Hao, N. Lane, X. Liu and T. Moscibroda. Cost-aware Compressive Sensing for Networked Sensing System. *Microsoft Tech Report*. 2015.
- [32] S. Yang and M. Gerla. Energy-efficient accelerometer data transfer for human body movement studies. In SUTC ’10, pages 304–311, 2010.
- [33] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang. A compressive sensing approach to urban traffic estimation with probe vehicles. *Mobile Computing, IEEE Trans. on*, 12(11):2289–2302, 2013.

APPENDIX

Proof of the Theorem 1.

PROOF. Let $\eta = 2 - \sigma(\mathbf{F}_\Omega)$, $\epsilon_{4k} = 2exp \left[-\frac{[\delta - (k-1)/(n-1)]^2 m^\eta}{8k} \right]$, and $\delta_{4k} = \frac{1}{2}$. In Theorem 1, assuming that all the constraints are satisfied, then we have

1. If $\sigma(\mathbf{F}_\Omega) < \frac{3}{2}$, we know $0.5 < \eta$

2. If $\sigma(\mathbf{F}_\Omega) < 2 - \log_m(4ck \log n)$, we have $m > \left(\frac{c k \log n}{\delta_{4k}^2} \right)^{1/\eta}$

3. If $\sigma(\mathbf{F}_\Omega) < 2 - \log_m \left[678.19k \left(\frac{n-1}{n-8k+1} \right)^2 \right]$, we have

$$\epsilon_{4k} = 2exp \left[-\frac{[\delta_{4k} - (k-1)/(n-1)]^2 m^\eta}{8k} \right] < 0.01$$

Finally from $k < \frac{n}{8}$, we have $4k < 1 + (n-1)\delta_{4k}$. According Lemma 1, the sensing matrix $\frac{1}{m}\mathbf{F}_\Omega$ with probability at least 0.99 satisfies RIP of order $4k$ with restricted isometry constant $\delta_{4k} = 0.5$. By applying ℓ_1 -min recovery algorithm, the theorem holds. \square