Learning Compressive Sensing Models for Big Spatio-Temporal Data*

Dongeun Lee† Jaesik Choi‡

Abstract

Sensing devices including mobile phones and biomedical sensors generate massive amounts of spatio-temporal data. Compressive sensing (CS) can significantly reduce energy and resource consumption by shifting the complexity burden of encoding process to the decoder. CS reconstructs the compressed signals exactly with overwhelming probability when incoming data can be sparsely represented with a fixed number of components, which is one of drawbacks of CS frameworks because a real-world signal in general cannot be represented with the fixed number of components. We present the first CS framework that handles signals without the fixed sparsity assumption by incorporating the distribution of the number of principal components included in the signal recovery, which we show is naturally represented by the gamma distribution. This allows an analytic derivation of total error in our spatio-temporal Low Complexity Sampling (LCS). We show that LCS requires shorter compressed signals than existing CS frameworks to bound the same amount of error. Experiments with real-world sensor data also demonstrate that LCS outperforms existing CS frameworks.

1 Introduction.

Various sensing devices including mobile phones and biomedical sensors are essential nowadays. Individually operating sensors usually form correlated sensor networks in large scale. Thus, these sensors generate continuous flows of *big sensing data* that pose important challenges: how to sense and transmit massive spatio-temporal data in efficient manner.

Many conventional distributed sensing schemes process input signals in the sensing devices to reduce the burden of network transmission [13], [14]. However, these conventional schemes are not well suited for resource limited sensing devices because of excessive energy and resource consumption.

Compressive sensing (CS) sheds light on this problem by shifting the complexity burden of encoding process to the decoder [3], [10]. CS enables to compress large amounts of inputs signals without much energy consumption. Recent advances in CS reduce this computational burden even further by *random sampling*, so that CS schemes are successfully applied to large-scale sensor networks [10], [11].

More specifically, CS reconstructs a compressed signal exactly with overwhelming probability when incoming data can be sparsely represented (i.e., a small number of principal components). In other words, the probability of recovery failure, where CS fails to reconstruct the exact input signal, can be bounded when a compressed signal has at least a predefined length.

However, existing CS frameworks are built based on a strong assumption which says that incoming data can be sparsely represented by a *fixed* number of principal components. This assumption does not hold in reality where the number of principal components cannot be determined. Existing CS frameworks consider that the reconstruction would fail, when an input signal has more principal components (denser) than the predefined threshold.

The main contribution of this paper is that we relax the assumption of fixed sparsity during signal recovery, and instead view the number of principal components included in the signal recovery as varying quantity. We present Low Complexity Sampling (LCS) that can efficiently sense and transmit big spatio-temporal data. LCS requires shorter compressed signals with given fidelity (better coding efficiency). We report that the number of principal components included in signal recovery is naturally represented by the *gamma distribution* in practice, which allows us an analytic derivation to compute the expected amount of error in LCS.

To our knowledge, this is the first CS framework to model the distribution of signal sparsity beyond the fixed threshold assumption. An error analysis with the gamma distribution enables our framework to learn the number of spatio-temporal measurements of input

*This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (NRF-2014R1A1A1002662), the NRF grant funded by the MSIP (NRF-2014M2A8A2074096), and the 2013 Research Fund (1.130056.01) of UNIST (Ulsan National Institute of Science and Technology).

†School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan 689-798, Korea. eundong@unist.ac.kr

‡School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, Ulsan 689-798, Korea. jaesik@unist.ac.kr (corresponding author)

signals in LCS. We also show that the coding efficiency of LCS is superior to a state-of-the-art CS scheme by both asymptotic analysis and experimental results.

The rest of this paper is organized as follows. §2 reviews CS frameworks. §3 introduces how LCS operates. In §4, we provide our main contribution, a new CS framework that incorporates the distribution of the number of principal components, which can eventually learn the number of spatio-temporal measurements. §5 exhibits the performance comparison of our framework compared to existing CS frameworks, followed by concluding remarks in §6.

2 Background: Compressive Sensing.

Compressive sensing or compressed sampling (CS) is an ideal option for mitigating computational complexity by shifting the complexity burden to the decoder where an original signal is estimated in best-effort manner [3], [7].

2.1 Compressing Signal During Acquisition. Conventional compression schemes obtain the compact representation of an original signal by encoding the significant coefficients because most signals can be represented with only a few components using approximations of Karhunen-Loève transform (e.g., the discrete cosine transform (DCT) and the wavelet transform [16]).

In CS, a signal is projected onto random vectors whose cardinality is far below the dimension of the signal. As an example, a signal $\mathbf{x} \in \mathbb{R}^N$ is compactly represented with a few orthogonal basis $\mathbf{\Psi}$ having large coefficients and many small coefficients close to zero as follows:

$$\mathbf{x} = \mathbf{\Psi}\mathbf{s}, \tag{2.1}$$

where $\mathbf{s} \in \mathbb{R}^N$ is the vector of transformed coefficients with a few significant coefficients.

In (2.1), $\mathbf{\Psi}$ is any orthogonal basis that makes $\mathbf{x}$ sparse in transform domain such as the DCT and wavelet transform domains. The signal $\mathbf{x}$ is called K-sparse if it is a linear combination of only $K \ll N$ basis vectors such that $\sum_{i=1}^{K} s_{n_i} \boldsymbol{\psi}_{n_i}$, where $\{n_1, \ldots, n_K\} \subset \{1, \ldots, N\}$; s_{n_i} is a coefficient in $\mathbf{s}$; and $\boldsymbol{\psi}_{n_i}$ is a column of $\mathbf{\Psi}$. Note that a *real world signal* in general is not exactly K-sparse; rather it can be closely approximated with K basis vectors.

CS projects $\mathbf{x}$ onto a random sensing basis $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ as follows ($M<N$):

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s}, \tag{2.2}$$

where $\mathbf{\Phi}$ is generally constructed by sampling independent identically distributed (i.i.d.) entries from the Gaussian or other sub-Gaussian distributions whose moment-generating function is bounded by that of the Gaussian (e.g., Rademacher/Bernoulli distribution).

The system shown in (2.2) is ill-posed as the number of equations M is smaller than the number of variables N: there are infinitely many $\mathbf{x}$'s that satisfy $\mathbf{y} = \mathbf{\Phi}\mathbf{x}$. Nevertheless this system can be solved with overwhelming probability provided that $\mathbf{s}$ is sparse and M is large enough such that $M = O(K \log(N/K))$ in the case of Gaussian and sub-Gaussian sensing matrices.

2.2 Recovery of Signal. A signal recovery algorithm takes measurements $\mathbf{y} \in \mathbb{R}^M$, a random sensing matrix $\mathbf{\Phi}$, and the sparsifying basis $\mathbf{\Psi}$.[1] The sensing matrix $\mathbf{\Phi}$ and sparsifying basis $\mathbf{\Psi}$ are assumed to be known to the decoder.

The signal recovery algorithm then recovers $\mathbf{s}$ knowing that $\mathbf{s}$ is sparse. Once we recover $\mathbf{s}$, the original signal $\mathbf{x}$ can be recovered through (2.1). It has been shown that the following linear program gives an accurate reconstruction of $\mathbf{s}$:

$$\operatorname{argmin} \|\tilde{\mathbf{s}}\|_1 \quad \text{subject to} \quad \mathbf{\Phi}\mathbf{\Psi}\tilde{\mathbf{s}} = \mathbf{y}. \tag{2.3}$$

There are many efficient algorithms that solve the optimization problem in (2.3) under categories of optimization methods, greedy methods, and thresholding-based methods [10].

2.3 Noisy Signal Recovery. Suppose $\mathbf{y}$ were corrupted with a noise $\mathbf{z} \in \mathbb{R}^M$ that is a stochastic or deterministic unknown error term, which could be from various sources such as communication and quantization. The corrupted $\hat{\mathbf{y}}$ can be represented as

$$\hat{\mathbf{y}} = \mathbf{\Phi}\mathbf{\Psi}\mathbf{s} + \mathbf{z}. \tag{2.4}$$

It has been shown that (2.4) can be solved using the following minimization problem with relaxed constraints for reconstruction:

$$\operatorname{argmin} \|\tilde{\mathbf{s}}\|_1 \quad \text{subject to} \quad \|\mathbf{\Phi}\mathbf{\Psi}(\mathbf{s} - \tilde{\mathbf{s}}) + \mathbf{z}\|_2 \leq \eta\sqrt{M}, \tag{2.5}$$

where $\eta\sqrt{M}$ bounds the amount of noise in the signal. The problem (2.5) is called a quadratically constrained minimization problem or LASSO (Least Absolute Shrinkage and Selection Operator), which can also be solved using various efficient algorithms similar to the recovery of noiseless signal [10].

[1] In a typical setup, the only information an encoder always has to send is $\mathbf{y}$. The sensing matrix $\mathbf{\Phi}$ can be explicitly sent [14] or reconstructed using meta information such as the seed of pseudorandom number generator [9], [12], depending on application.

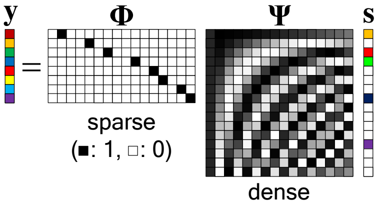

Figure 1: Random sampling of a signal in (2.2).

3 Low Complexity Sampling.

The sensing/sampling paradigm explained in §2 can be applied to signals that can be represented in one-dimensional vectors, $\mathbf{x}$. These vectors not only correspond to time-series data in temporal dimension, but also to data in spatial dimension at a specific time instant, in which case two-dimensional data can be vectorized into an one dimension.

3.1 Handling Spatio-Temporal Dimension. Recently, the compressive sensing (CS) technique has been applied to spatial and temporal dimensions. One straightforward extension is to vectorize the spatial and temporal dimensions. However, the vectorization easily infringe the sparsity of signals. Thus, CS methods for correlaed spatio-temporal signals were introduced [4], [5], [9]. These schemes improved the coding efficiency with additional computational and communicational overheads. Low Complexity Sampling (LCS) is an alternative to reduce these burdens when spatial data can be sparsely represented for each selected time instant [11].

LCS utilizes more efficient sensing mechanism based on *random sampling* whose computational time complexity is constant [10]. The random sampling scheme is based on the fact that it is possible to construct $\mathbf{\Phi}$ in (2.2) from a random selection of rows from the identity matrix $\mathbf{I}$, which is equivalent to the random sampling of coefficients in $\mathbf{x}$. Note that the sparsifying basis $\mathbf{\Psi}$ should be *incoherent*[2] with $\mathbf{I}$ such as the DCT and wavelet transform bases, in order not to violate the condition for the successful recovery of an original signal [7], [10]. The random sampling of a signal in the CS setup is illustrated in Fig. 1. Here, the number of required measurements M is somewhat larger than the case of Gaussian and sub-Gaussian matrices, that is, $M = O(K \log N)$.

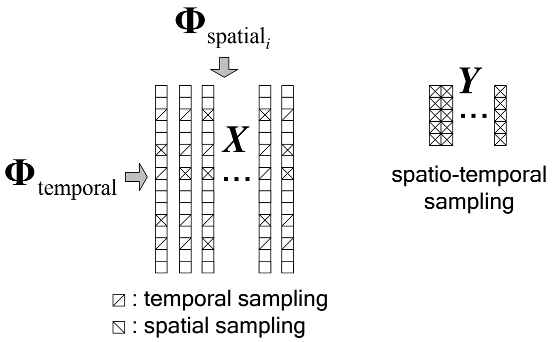

Figure 2: Low Complexity Sampling (LCS) of spatio-temporal data.

3.2 Signal Acquisition. LCS randomly samples an original signal in both spatial and temporal dimensions. First, each time-series data in temporal dimension is randomly sampled using the same indices shared across spatial dimension, which is equivalent to using the same random sensing matrix $\mathbf{\Phi}_{\text{temporal}}$ across spatial dimension for the same time frame. This sampling reduces the lengths of original time-series data.

Next, the random sampling is performed in the spatial dimension: the group of randomly sampled coefficients in the temporal dimension is sampled again in the spatial dimension, which is illustrated in Fig. 2. Therefore, when used in a distributed environment, each device needs to sample and transmit only coefficients that are selected in both spatial and temporal dimensions, as shown in Fig. 2.

The LCS framework can be formally represented as follows:

$$\boldsymbol{Y}_{\text{temporal}} = \mathbf{\Phi}_{\text{temporal}}\boldsymbol{X}, \tag{3.6}$$

where $\boldsymbol{X} = [\mathbf{x}_1\mathbf{x}_2 \cdots \mathbf{x}_J]$ is an $N \times J$ matrix having temporal vectors $\mathbf{x}_i \in \mathbb{R}^N$ of J distributed devices as columns; $\mathbf{\Phi}_{\text{temporal}} \in \mathbb{R}^{M \times N}$ is the temporal random sampling matrix shared across the spatial dimension with $M < N$; $\boldsymbol{Y}_{\text{temporal}} \in \mathbb{R}^{M \times J}$ is a half-encoded matrix in temporal dimension.

Now $\boldsymbol{Y}_{\text{temporal}}$ has vectors $\mathbf{y}_{\text{temporal}_1}^{\mathrm{T}}, \ldots, \mathbf{y}_{\text{temporal}_M}^{\mathrm{T}}$ as rows. Then each row $\mathbf{y}_{\text{temporal}_i}^{\mathrm{T}} \in \mathbb{R}^J$ in $\boldsymbol{Y}_{\text{temporal}}$ is randomly sampled in the spatial dimension, which is given by

$$\mathbf{y}_i = \mathbf{\Phi}_{\text{spatial}_i}\mathbf{y}_{\text{temporal}_i}, \tag{3.7}$$

where $\mathbf{\Phi}_{\text{spatial}_i} \in \mathbb{R}^{I \times J}$ is a spatial random sampling matrix for each $\mathbf{y}_{\text{temporal}_i}$ that yields a final measurement vector $\mathbf{y}_i \in \mathbb{R}^I$ $(I < J)$. Using (3.7), we can construct the final measurement matrix $\boldsymbol{Y} \in \mathbb{R}^{M \times I}$ shown in Fig. 2 by putting together $\mathbf{y}_i^{\mathrm{T}}$'s as rows of $\boldsymbol{Y}$, where M

[2] The two bases $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are (maximally) incoherent when the largest correlation between any two elements of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ is $1/\sqrt{N}$ where N is the order of two square matrices.

is the number of measurements in the temporal dimension and I the number of measurements in the spatial dimension.

The rationale behind using the same random sampling indices in the temporal dimension is to maximize the spatial correlation of coefficients across different devices: the spatial correlation is stronger at the same time instant than with different time instants under some mild conditions as in [17], which is proved in the following theorem.

Theorem 3.1. *The spatial correlation of a signal across devices is maximized with spatial data taken at the same time instant, when spatial and temporal correlations are nonnegative and decrease monotonically with a distance between devices and a time lag.*

Proof. We first assume the wide sense stationarity of a random variable X_n that denotes each spatial datum and define the correlation matrix $\boldsymbol{R} \in \mathbb{R}^{J \times J}$ whose (i,j)th entry $\boldsymbol{R}_{i,j}$ is given by

$$\boldsymbol{R}_{i,j} = \mathbf{E}[X_n X_{n+|i-j|}]. \tag{3.8}$$

Then we want to show the following inequality

$$\boldsymbol{R}^{\star}_{i,j} \geq \boldsymbol{R}_{i,j} \qquad \forall i,j \tag{3.9}$$

holds for any $\boldsymbol{R}$ if and only if

$$\boldsymbol{R}^{\star}_{i,j} = \frac{1}{J-|i-j|} \sum_{n=1}^{J-|i-j|} x_n x_{n+|i-j|}{}^{3} \tag{3.10}$$

with $x_n \in \{\boldsymbol{X}_{i_1,1}, \ldots, \boldsymbol{X}_{i_J,J}\}$ where $i_1 = i_2 = \cdots = i_J$.

From the condition [17], one can model correlations using an *exponential function* such as $R(\tau) = \mathrm{e}^{-|\tau|/\theta}$ where τ is the distance or the time lag; $\theta > 0$ is a scale parameter that determines the shape of the correlation function. (Therefore, without loss of generality, we here assume a normalized version of the correlation model.) If the distance is denoted by τ_{spatial} and the time lag by τ_{temporal}, then $\tau_{\text{spatial}} = l - j$ and $\tau_{\text{temporal}} = k - i$ when we compare two entries $\boldsymbol{X}_{i,j}$ and $\boldsymbol{X}_{k,l}$.

In general, the spatial and temporal correlations are treated separately and the effect of one on the other is ignored [17]. Since we assume correlation models both have the monotonically decreasing property, it is reasonable to think that a joint spatio-temporal correlation model $R(\tau_{\text{spatial}}, \tau_{\text{temporal}})$ decreases monotonically with both τ_{spatial} and τ_{temporal}, such as in $R(\tau_{\text{spatial}}, \tau_{\text{temporal}}) = \mathrm{e}^{-(|\tau_{\text{spatial}}|/\theta_{\text{spatial}} + |\tau_{\text{temporal}}|/\theta_{\text{temporal}})}$ where θ_{spatial} is the scale parameter for the distance and θ_{temporal} for the time lag. Hence, the necessary and sufficient condition for (3.9) to hold is $i_1 = i_2 = \cdots = i_J$, which means x_n is taken at the same time instant across spatial data.

Strong correlation improves the coding efficiency, which can be explained with the help of information theory: the joint entropy rate (i.e., compressed size) of two or more random variables is always less than the sum of their individual rates provided that they are dependent on each other [8]. If random variables are correlated, we can say they are dependent on each other, which leads to a reduced joint entropy rate. In the CS setup and eventually LCS, the correlation and the dependency are present in the form of sparse representations.

This framework is especially useful in a distributed sampling context thanks to the opt-in and opt-out nature of participating devices. In a distributed sensing, each device may want to participate in or not depending on its remaining energy [13] or its willingness to volunteer in the context of *participatory sensing* [6]. If these choices are made random from a holistic perspective, large scale sampling tasks can be efficiently performed.

Regarding how to generate random numbers used for the random sampling of spatio-temporal data, popular approaches are to use pseudorandom numbers [9], [12]. These random numbers should be synchronized between the encoder and the decoder, in order to ensure the correct recovery of an original signal as explained in §2.2. Randomness can also be improved via periodically updating random seeds between the encoder and the decoder. Note that LCS only needs to store random indices of coefficients in spatial and temporal dimensions, instead of storing entries for the random sensing matrix $\mathbf{\Phi}$.

Furthermore, the random indices for the spatial sampling need not be explicitly synchronized between the encoder and the decoder if each device in a distributed environment determines whether to transmit or not at every time instant that corresponds to the temporal sampling point. This is possible because the collection point (decoder) can recognize the spatial index required for the reconstruction when it receives sampled data from each device.

We finally describe the algorithm of LCS. In particular, the operation of each device participating in a distributed sampling task can be represented as the following algorithm. (We assume that the random indices for the spatial sampling are explicitly synchronized between the encoder and the decoder.)

Require: N, temporal random index set $T =$

[3] This is a general method of computing the correlation matrix [16].

$\{n_1, \ldots, n_M\} \subset \{1, \ldots, N\}$[4], and spatial random index vector $\mathbf{s} \in \mathbb{R}^M$ ($s_i \in \{0, 1\}$) {$\mathbf{s}$ is the translated version of the spatial random indices from individual devices' perspective}

```
loop
  $j \leftarrow 1$
  for $i = 1$ to $N$ do
    if $i \in T$ then
      if $s_j = 1$ then
        transmit data
      end if
      $j \leftarrow j + 1$
    end if
  end for
end loop
```

3.3 Signal Recovery. The collection point takes a random sample measurement matrix as described in Fig. 2. Note that the recovery problem in this case is different from conventional CS recovery in the sense that we are dealing with a measurement *matrix*, not a measurement *vector*. Thus we need to decode the measurement matrix $\boldsymbol{Y} \in \mathbb{R}^{M \times I}$.

First, each row of $\boldsymbol{Y}$ is decoded following the recovery procedure explained in §2.2. Specifically, the solution $\mathbf{s}^\star$ to (2.3) obeys

$$\|\mathbf{s}^\star - \mathbf{s}\|_2 \leq C_1 \cdot \|\mathbf{s} - \mathbf{s}_K\|_1 \tag{3.11}$$

for some constant $C_1 > 0$, where $\mathbf{s}_K$ is the vector $\mathbf{s}$ with all but the largest K components set to 0: the quality of recovered signal is proportional to that of the K most significant pieces of information. We get progressively better results as we compute more measurements I, since $I = O(K \log J)$ [7]. Therefore, $\boldsymbol{\Psi}\mathbf{s}^\star \in \mathbb{R}^J$ also makes progress on its quality as I increases. (The error bound follows (3.11) as well if $\boldsymbol{\Psi}$ is an orthogonal matrix, which is usually the case.)

We then have a half-decoded matrix $\widehat{\boldsymbol{Y}}_{\text{temporal}}$ whose size is $M \times J$. Each column of the half-decoded matrix is decoded once more to obtain the full-decoded matrix $\widehat{\boldsymbol{X}} \in \mathbb{R}^{N \times J}$. In contrast to the case of decoding each row of $\boldsymbol{Y}$, decoding each column follows the recovery procedure explained in §2.3, because error occurs during the recovery of spatially sampled coefficients following the error bound in (3.11).

In particular, the solution $\mathbf{s}^\star$ to (2.5) obeys the following reconstruction error bound:

$$\|\mathbf{s}^\star - \mathbf{s}\|_2 \leq C_1 \cdot \|\mathbf{s} - \mathbf{s}_K\|_1 + C_2 \cdot \sqrt{K}\eta, \tag{3.12}$$

where $C_2 > 0$ is another constant for the additional term in the new error bound. Thus, (3.12) accounts for not only the measurement error due to an insufficient M, but the measurement error carried over from the previous recovery stage, which is explained by η in (2.5).

4 Learning the Number of Spatio-Temporal Measurements.

In §3.3, the recovery of the measurement matrix $\boldsymbol{Y}$ in (3.11) and (3.12) involved error to some extent. We here investigate the amount of error occurring during the recovery procedure in *expected value* sense. In fact, reconstruction in compressive sensing (CS) is closely related to a probabilistic concept. For instance, when we say an exact recovery of a K-sparse signal is achievable with overwhelming probability, it implies there is also the chance of recovery not being exact.

Specifically, the number of required measurements $M = O(K \log N)$ in the random sampling can be detailed as follows [10]:

$$M \geq C \cdot K \ln(N) \ln(\epsilon^{-1}) \tag{4.13}$$

for some constant $C > 0$, where $\epsilon \in (0, 1)$ denotes the probability of an *inexact* recovery of a K-sparse signal. Then we can express (4.13) with regard to ϵ, which is given by

$$\epsilon = \exp\left(-\frac{M}{C \cdot \ln(N) K}\right). \tag{4.14}$$

Most existing CS frameworks assume that K is already known, which is not true in practice. Thus, we pose a new learning problem for Low Complexity Sampling (LCS) and provide a new error analysis. In our scenario where a signal is not exactly K-sparse, this can be interpreted as varying numbers of K largest components during signal recovery in (3.11). Since a fixed K does not exist here, one may think if a signal recovery fails with a given K, then K can be lowered to decrease ϵ and the recovery eventually succeeds.

4.1 Sparsity of Stochastic Nature. With varying K, we regard it as a random variable and consider its probability density function (pdf). One possible method for obtaining this pdf is to derive it from (4.14) by viewing *the probability of failure* ϵ as the cumulative distribution function (CDF) of K.

Definition 4.1. The pdf of K components included in signal recovery, which is derived from the probability of failure in the CS framework, is given by

$$f_K^{CS}(k) := \frac{M}{C \cdot \ln(N)} \cdot \exp\left(-\frac{M}{C \cdot \ln(N) k}\right) \frac{1}{k^2}. \tag{4.15}$$

However, this pdf fails to capture an actual distribution. We instead found empirically K followed

[4] This can be generated by a random permutation.

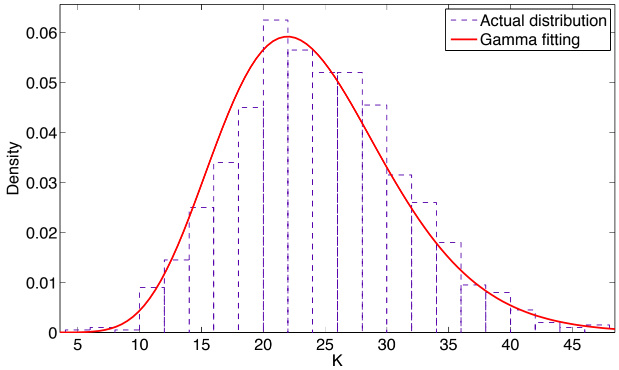

Figure 3: Distribution of K fitted with a gamma distribution Gamma$(11.78, 0.49)$, using the *maximum-likelihood estimation*. Histogram was obtained with 1,000 experiments over the same signal using (3.11). The constant C_1 in (3.11) can be inferred from other experiments with exact K terms that calculate actual ϵ over varying K's. We find K that yields $\epsilon = 0.5$ and set it to be the mean of $f_K(k)$.

the *gamma distribution* with its mean controlled by the number of measurements M in (4.13) such that $\mathbf{E}[K] = C' \cdot M$. (Note that $\ln(\epsilon^{-1})$ is gone and $1/(C \cdot \ln(N)\ln(\epsilon^{-1}))$ is absorbed into C'.) Fig. 3 illustrates $f_K(k) = \text{Gamma}(\alpha_K, \beta_K)$ follows an actual distribution properly where $f_K(k)$ is drawn over a histogram of K's obtained with experiments.

In fact, the problem of using $f_K^{CS}(k)$ as the pdf of K components included in signal recovery lies at its fat tail. We can show that $f_K^{CS}(k)$ has a fatter tail than the gamma distribution has by the following lemma.

Lemma 4.1. *The pdf $f_K^{CS}(k)$ is a fat-tailed distribution.*

Proof. It is sufficient to show that $f_K^{CS}(k)$ is asymptotically equivalent to a power-law decay such that

$$f_K^{CS}(k) \sim k^{-\alpha} \text{ as } k \to \infty, \quad \alpha > 1. \tag{4.16}$$

Therefore, we show the following inequality

$$\frac{M}{C \cdot \ln(N)} \cdot \exp\left(-\frac{M}{C \cdot \ln(N)k}\right)\frac{1}{k^2} > k^{-\alpha} \tag{4.17}$$

holds for some $\alpha > 1$ as $k \to \infty$. Taking logarithms on both sides, we obtain

$$\alpha > \frac{M}{C \cdot \ln(N)\ln(k)k} - \frac{\ln(M/(C \cdot \ln(N)))}{\ln(k)} + 2, \tag{4.18}$$

which turns into $\alpha > 2$ as $k \to \infty$. Hence we can argue that $f_K^{CS}(k)$ is a fat-tailed distribution.

Corollary 4.1. *The tail of $f_K^{CS}(k)$ is fatter than the tail of gamma distribution.*

Proof. It is known that the gamma distribution is not a fat-tailed distribution. Hence, we can say $f_K^{CS}(k)$ has a fatter tail than the gamma distribution has.

4.2 Learning in Spatial Domain. When each row of the measurement matrix $\boldsymbol{Y}$ is decoded in the first stage, inevitable error due to the inexact sparsity occurs according to (3.11). In particular, the best K-term approximation $\|\mathbf{s} - \mathbf{s}_K\|_1$ in (3.11) is known to be bounded as follows [10]:

$$\|\mathbf{s} - \mathbf{s}_K\|_1 \leq \frac{1}{4K}\|\mathbf{s}\|_{0.5}. \tag{4.19}$$

Here the quantity $\|\mathbf{s}\|_{0.5}$ can also vary across rows. Thus we again regard it as a random variable and denote it by G_S, whose arbitrary pdf is also denoted by $f_{G_\text{S}}(g_\text{S})$. In addition, we use K_S to represent the number of significant pieces of information included in the recovery of each row, which is controlled by the number of measurements I.

We can then analyze the error of each decoded row E_S using (3.11):

$$\begin{aligned} E_\text{S} &= \iint\limits_{k_\text{S}\, g_\text{S}} f_{K_\text{S}}(k_\text{S}) f_{G_\text{S}}(g_\text{S}) \cdot \frac{C_1}{4k_\text{S}} g_\text{S}\, \mathrm{d}g_\text{S}\, \mathrm{d}k_\text{S} \\ &= \frac{C_1}{4}\beta_{K_\text{S}} \mathrm{B}(\alpha_{K_\text{S}} - 1, 1) \int_{g_\text{S}} g_\text{S} f_{G_\text{S}}(g_\text{S})\, \mathrm{d}g_\text{S}, \end{aligned} \tag{4.20}$$

where $\mathrm{B}(\cdot,\cdot)$ is the beta function; $\int_{g_\text{S}} g_\text{S} f_{G_\text{S}}(g_\text{S})\, \mathrm{d}g_\text{S}$ is simply the mean of $f_{G_\text{S}}(g_\text{S})$ that can be easily found according to a specified pdf. Note that the error of each decoded row E_S can be rescaled to yield the entry-wise error estimate by $E_\text{S}/\sqrt{J}$.[5]

4.3 Learning in Temporal Domain. The first stage decoding produces the half-decoded matrix $\widehat{\boldsymbol{Y}}_\text{temporal}$ whose size is $M \times J$. In the second stage, each column of the half-decoded matrix is decoded. This time we should also consider the measurement error $E_\text{S}/\sqrt{J}$ carried over from the first stage, which can be embodied in (3.12).

In particular, we can analyze the error of each decoded column E_T using (3.12). Once again, $\|\mathbf{s}\|_{0.5}$ in (4.19) varies across columns and is denoted by a random variable G_T, whose arbitrary pdf is represented by $f_{G_\text{T}}(g_\text{T})$. In addition, the number of significant pieces of information included in the recovery of each

[5] A division by $\sqrt{J}$ is natural as we are dealing with ℓ_2 error.

column is represented by K_{T}, which is controlled by the number of measurements M. Then the error is given by

$$
\begin{aligned}
E_{\mathrm{T}} &= \iint\limits_{k_{\mathrm{T}} g_{\mathrm{T}}} f_{K_{\mathrm{T}}}(k_{\mathrm{T}}) f_{G_{\mathrm{T}}}(g_{\mathrm{T}}) \\
&\quad \cdot \left(\frac{C_1}{4k_{\mathrm{T}}} g_{\mathrm{T}} + \frac{C_2 E_{\mathrm{S}}}{\sqrt{J}} \sqrt{k_{\mathrm{T}}} \right) \mathrm{d}g_{\mathrm{T}}\, \mathrm{d}k_{\mathrm{T}} \\
&= \frac{C_1}{4} \beta_{K_{\mathrm{T}}} \mathrm{B}(\alpha_{K_{\mathrm{T}}} - 1, 1) \int_{g_{\mathrm{T}}} g_{\mathrm{T}} f_{G_{\mathrm{T}}}(g_{\mathrm{T}})\, \mathrm{d}g_{\mathrm{T}} \\
&\quad + \frac{C_2 E_{\mathrm{S}}}{\sqrt{J \beta_{K_{\mathrm{T}}}}} \frac{\Gamma(1/2)}{\mathrm{B}(\alpha_{K_{\mathrm{T}}}, 1/2)}, \qquad (4.21)
\end{aligned}
$$

where $\Gamma(\cdot)$ is the gamma function; $\int_{g_{\mathrm{T}}} g_{\mathrm{T}} f_{G_{\mathrm{T}}}(g_{\mathrm{T}})\, \mathrm{d}g_{\mathrm{T}}$ is the mean of $f_{G_{\mathrm{T}}}(g_{\mathrm{T}})$ that can be found according to a specified pdf.

4.4 Total Error. Now that we are provided with the error of each decoded column E_{T}, we can rescale it to find the total error by $\sqrt{J} E_{\mathrm{T}}$, which is equivalent to the expected value of the Frobenius norm of the error matrix $\mathbf{E}[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F]$ that accounts for the difference between the original signal $\boldsymbol{X}$ and the reconstructed signal $\widehat{\boldsymbol{X}}$.

We then consider the following convex optimization problem that minimizes the total error with a given measurement budget R_0:

(4.22)

$$
\underset{\{I,M\}}{\operatorname{argmin}} \mathbf{E}\left[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F \right] \qquad \text{subject to} \qquad IM \leq R_0.
$$

For a fixed I, the objective function $\mathbf{E}[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F]$ in (4.22) has a global minimum due to the term $C_2 \cdot \sqrt{K}\eta$ in (3.12): increasing M does not warrant a consistent decrease in total error.

In order to examine more closely, we calculate the gradient of $\mathbf{E}[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F]$ with respect to I and M so as to investigate the shape of the objective function. Specifically in (4.20) and (4.21), $\beta_{K_{\mathrm{S}}}$ and $\beta_{K_{\mathrm{T}}}$ can be replaced by $\alpha_{K_{\mathrm{S}}}/(C' \cdot I)$ and $\alpha_{K_{\mathrm{T}}}/(C' \cdot M)$ respectively[6] since the mean of gamma distribution is α_K/β_K. Then $\partial \mathbf{E}[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F]/\partial I < 0$, which means increasing I always helps reducing the total error. On the contrary, we can find M that makes $\partial \mathbf{E}[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F]/\partial M = 0$ where $M \sim I^{2/3}$.

4.5 Error Estimation Accuracy. We now examine the accuracy of our error estimation compared with the error estimation using $f_K^{CS}(k)$. We can prove our error estimation with the gamma distribution yields more accurate results than using $f_K^{CS}(k)$ by the following theorem.

Theorem 4.1. *The accuracy of error analysis for LCS is improved by using a proper distribution such as the gamma distribution.*

Proof. We show the error of each decoded row E_{S} in (4.20) is smaller than E_{S} computed with $f_{K_{\mathrm{S}}}^{CS}(k_{\mathrm{S}})$ if $\alpha_{K_{\mathrm{S}}} > \alpha_{K_{\mathrm{S}}}^{\star}$ for some threshold $\alpha_{K_{\mathrm{S}}}^{\star} > 0$. For this, we consider the following ratio r of E_{S} with $f_{K_{\mathrm{S}}}^{CS}(k_{\mathrm{S}})$ to E_{S} in (4.20):

$$
\begin{aligned}
r &= \frac{\int_0^{\infty} \frac{I}{C \cdot \ln(J)} \cdot \exp(-\frac{I}{C \cdot \ln(J) k_{\mathrm{S}}}) \frac{1}{{k_{\mathrm{S}}}^2} \cdot \frac{C_1}{4k_{\mathrm{S}}}\, \mathrm{d}k_{\mathrm{S}}}{\frac{C_1}{4} \beta_{K_{\mathrm{S}}} \mathrm{B}(\alpha_{K_{\mathrm{S}}} - 1, 1)} \\
&= \frac{\frac{C_1 \cdot C \cdot \ln(J)}{4I}}{\frac{C_1 \cdot C \cdot \ln(J) \ln(\epsilon^{-1}) \alpha_{K_{\mathrm{S}}} \mathrm{B}(\alpha_{K_{\mathrm{S}}} - 1, 1)}{4I}} \\
&= \frac{\alpha_{K_{\mathrm{S}}} - 1}{\ln(\epsilon^{-1}) \alpha_{K_{\mathrm{S}}}}. \qquad (4.23)
\end{aligned}
$$

In (4.23), $r > 1$ when $\alpha_{K_{\mathrm{S}}} > 1/(1 - \ln(\epsilon^{-1}))$. Thus, $\alpha_{K_{\mathrm{S}}}^{\star} = 1/(1 - \ln(\epsilon^{-1}))$, where we set $\epsilon = 0.5$ as shown in Fig. 3. It is also clear that the error of each decoded column E_{T} in (4.21) suffers inaccuracy with the pdf $f_{K_{\mathrm{T}}}^{CS}(k_{\mathrm{T}})$. In fact, both of these inaccuracies are attributed to the fat tail of $f_K^{CS}(k)$, as proven in Lemma 4.1.

In fact, this theorem can be generalized using distributions other than the gamma distribution that explain the actual distribution of K components included in signal recovery more suitably.

5 Performance Comparison.

The performance of compression schemes can be judged by comparing the coding efficiency that accounts for how compact data can be compressed with given fidelity. In other words, the scheme with the maximum coding efficiency among others is one that minimizes error between an original signal and a reconstructed signal with a given number of measurements.

Here we compare Low Complexity Sampling (LCS) with Model-Based CS (block sparse model), which is so far a state-of-the art compressive sensing (CS) scheme that takes account of joint correlation in spatio-temporal dimension [4], [5], [9]. Model-Based CS is built on a *joint sparsity model* in order to exploit the joint correlation inherent in the spatio-temporal dimension. Specifically, it utilizes *common sparse supports* among the group of signal vectors in temporal dimension: it assumes all data vectors in the temporal dimension measured by devices share the same K basis vectors. Thus the decoder has to select the union of individual supports to cover all of sparse supports from different devices, which makes the common K larger.

[6] Obviously, C' here is different from each other.

In order to maximize the joint correlation, Model-Based CS makes every device involved in measuring each other's data [4]. This aggressive participation leads to the maximum coding efficiency from *the perspective of the decoder* at the expense of high-complexity burdens on individual devices. In particular, individual measurements of devices are summed at the decoder to produce a complete measurement vector. On the contrary, individual measurements are separately treated in LCS and other CS schemes, which results in a low-complexity burden and even a *constant-time complexity* for LCS [11].

5.1 Asymptotic Analysis. We are particularly interested in the coding efficiency per the measurements of *individual devices*. To this end, we first analyze it asymptotically.

LCS has the total error $\mathbf{E}[\|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F] \sim \sqrt{J}/M + \sqrt{M}/I$. In contrast, Model-Based CS needs the number of measurements $M = O(JK + K\log(N/K))$ to yield error asymptotically equivalent to $1/\sqrt{K}$ (see Theorem 6 [4]). After rescaling them to directly compare with the total error of LCS per the same measurements of individual devices, we obtain the rescaled error in terms of I and M, which is $J/\sqrt{IM}$.

Using the fact that I and M are the same orders and $I < J$, we can show LCS has smaller total error per the same measurements of individual devices than Model-Based CS has. Since the total error is compared with the same numbers of measurements, we can say the coding efficiency of LCS is superior to that of Model-Based CS in terms of expenditure of individual devices.

5.2 Experimental Results. We also provide experimental results with real data sets. Experiments additionally include Spatial Random Sampling that performs the random sampling in spatial dimension. Sensing in the spatial dimension is the most popular approach that utilizes CS, hence adopted here for reference [2], [12], [14].

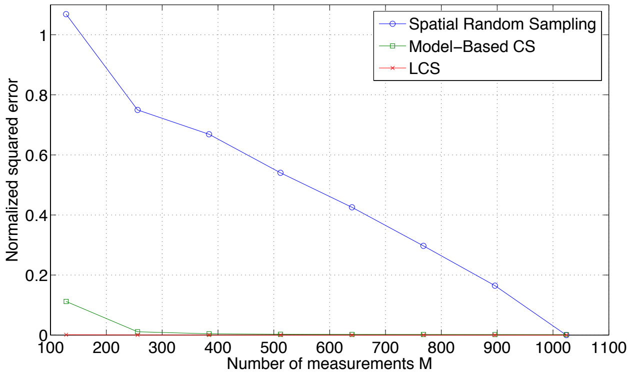

Figure 4: Normalized SSE comparison of ambient temperature signals from 32 sensors ($J = 32$) with the temporal data length $N = 1024$.

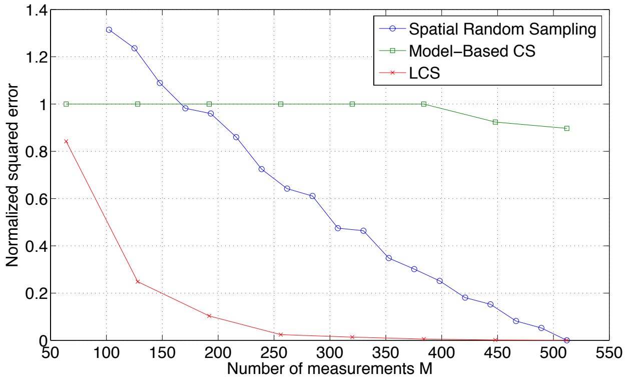

Figure 5: Normalized SSE comparison of luminosity signals from 45 sensors ($J = 45$) with the temporal data length $N = 512$.

Since we are interested in real-world signals, we employed environmental data sets obtained from wireless sensor network deployment scenarios [1], [15]. In particular, we employed three different sensor data types for our experiments: (i) ambient temperature (°C) [1]; (ii) luminosity (V) and (iii) battery level (V) [15]. These signals are not exactly K-sparse and instead approximated with K basis vectors. Specifically, we utilized the DCT throughout the experiments as the sparsifying basis $\boldsymbol{\Psi}$. As discussed in §3.3, recovered signal quality is improved (i.e., signal error is decreased) as more random measurements are received by the decoder, because the number of significant pieces of information included in signal recovery K is increased according to the number of measurements.

Figs. 4, 5, and 6 show the experimental results of three environmental data sets. We here consider the sum of squared error (SSE) normalized with respect to the norm of signals as the performance metric. All of the three schemes show decreasing normalized SSEs as the number of measurements per individual devices increases. However, how fast a normalized SSE drops depends on various schemes. In Fig. 4, Spatial Random Sampling shows an unsatisfactory result. This is mostly attributed to stronger intra-signal correlation inside each sensor device than inter-signal correlation between sensor devices: sole consideration of the spatial dimension fails to capture significant correlation in the temporal dimension. On the contrary, LCS and Model-Based CS both show superior results.

Whereas in Figs. 5 and 6, temporal data lengths are

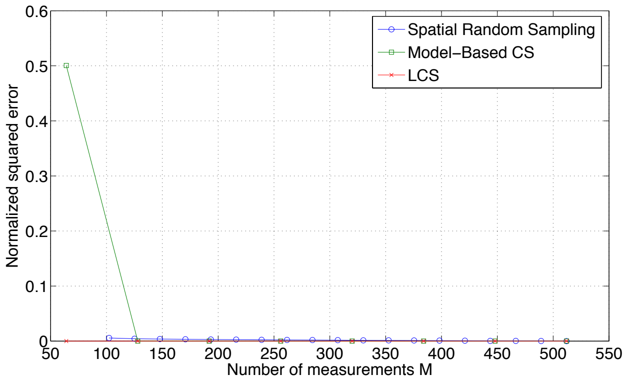

Figure 6: Normalized SSE comparison of battery level signals from 45 sensors ($J = 45$) with the temporal data length $N = 512$.

shorter and the numbers of distributed devices J's are slightly larger. These conditions mitigate the limitation of Spatial Random Sampling. Specifically in Fig. 6 where signals are heavily correlated (small entropy), the performance of Spatial Random Sampling is comparable to LCS.

In contrast, Model-Based CS suffers large normalized SSEs in Figs. 5 and 6. This coding inefficiency mainly comes from the increased J's. In addition, since the joint sparsity model in Model-Based CS utilizes common sparse supports among the group of signal vectors in temporal dimension, the decoder has to select the union of individual supports to cover all of sparse supports from different devices, whose cardinality (common K) increases as J increases. This makes the number of measurements insufficient for the decoder to recover the original signal. Contrasted with Spatial Random Sampling and Model-Based CS, LCS shows the minimum normalized SSEs across three different sensor data types.

6 Conclusion.

We have proposed Low Complexity Sampling (LCS) that can facilitate big spatio-temporal data sensing with low overheads. We presented the first CS framework that handles signals without the fixed sparsity assumption by viewing the number of principal components included in signal recovery as varying quantity. This quantity was shown to follow the gamma distribution that enabled an error analysis on LCS, from which the number of spatio-temporal measurements has been learned. Asymptotic analysis and experiments with real-world sensor data sets demonstrate that LCS outperforms existing CS frameworks.

References

[1] Sensorscope: Sensor networks for environmental monitoring. http://lcav.epfl.ch/op/edit/sensorscope-en

[2] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, *Compressive wireless sensing*, in Proc. Int'l Conf. Inf. Process. Sens. Netw. (IPSN '06), 2006, pp. 134–142.

[3] R. G. Baraniuk, *Compressive sensing [lecture notes]*, IEEE Signal Process. Mag., 24 (2007), pp. 118–121.

[4] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, *Model-based compressive sensing*, IEEE Trans. Inf. Theory, 56 (2010), pp. 1982–2001.

[5] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, *Distributed compressive sensing*, arXiv preprint arXiv:0901.3403, 2009.

[6] J. A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, *Participatory sensing*, Center for Embedded Network Sensing, 2006.

[7] E. J. Candès and M. B. Wakin, *An introduction to compressive sampling*, IEEE Signal Process. Mag., 25 (2008), pp. 21–30.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, 2nd ed., 2006.

[9] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, *Universal distributed sensing via random projections*, in Proc. Int'l Conf. Inf. Process. Sens. Netw. (IPSN '06), 2006, pp. 177–185.

[10] S. Foucart and H. Rauhut, *A Mathematical Introduction to Compressive Sensing*, Springer, 2013.

[11] D. Lee and J. Choi, *Low complexity sensing for big spatio-temporal data*, in Proc. Int'l Conf. Big Data (BigData '14), 2014, pp. 323–328.

[12] C. Luo, F. Wu, J. Sun, and C. W. Chen, *Compressive data gathering for large-scale wireless sensor networks*, in Proc. MobiCom '09, 2009, pp. 145–156.

[13] D. Noh, D. Lee, and H. Shin, *QoS-aware geographic routing for solar-powered wireless sensor networks*, IEICE Trans. Commun., 90 (2007), pp. 3373–3382.

[14] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, *On the interplay between routing and signal representation for compressive sensing in wireless sensor networks*, in Proc. Inf. Theory Appl. Workshop (ITA '09), 2009, pp. 206–215.

[15] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, *Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework*, IEEE Trans. Wireless Commun., 11 (2012), pp. 3447–3461.

[16] K. Sayood, *Introduction to Data Compression*, Morgan Kaufmann, 4th ed., 2012.

[17] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, *Spatio-temporal correlation: Theory and applications for wireless sensor networks*, Comput. Netw., 45 (2004), pp. 245–259.