Compressed Sensing & Network Monitoring

Jarvis Haupt, Waheed U. Bajwa, Michael Rabbat, & Robert Nowak

*Reprinted with permission of IEEE. Originally published in IEEE Signal Processing Magazine, pp 92-101, March 2008, J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, Compressed Sensing for Networked D*ata. (©2008 IEEE)

Introduction

Network monitoring and inference is an increasingly important component of intelligence gathering, from mapping the structure of the Internet, to discovering clandestine social networks, as well to information fusion in wireless sensor networks. Indeed, several international conferences are dedicated to the nascent field of network science. This article considers a particularly salient aspect of network science that revolves around large-scale distributed sources of data and their storage, transmission, and retrieval. The task of transmitting information from one point to another is a common and well-understood exercise. But the problem of efficiently sharing information from and among a vast number of distributed nodes remains a great challenge, primarily because we do not yet have well developed theories and tools for distributed signal processing, communications, and information theory in large-scale networked systems.

The problem is illustrated by a simple example. Consider a network of n nodes, each having a piece of information or data $x_j, j=1,...n$. These data could be files to be shared, or simply scalar values corresponding to node attributes or sensor measurements. Let us assume that each x_j is a scalar quantity for the sake of this illustration. Collectively these data $x=[x_1,...,x_n]^T$, arranged in a vector, are called *networked data* to emphasize both the distributed nature of the data and the fact that they may be shared over the underlying communications infrastructure of the network. The networked data vector may be very large; n may be a thousand, a million, or more. Thus, even the process of gathering x at a single point is daunting (requiring n communications at least). Yet this global sense of the networked data is crucial in applications ranging from network security to wireless sensing. Suppose, however, that it is possible to construct a highly compressed version of x, efficiently and in a decentralized fashion. This would offer many obvious benefits, provided that the compressed version could be processed to recover x to within a reasonable accuracy.

There are several decentralized compression strategies that could be utilized. One possibility is that the correlations between data at different nodes are known a priori. Then distributed source

coding techniques, such as Slepian-Wolf coding, can be used to design compression schemes without collaboration between nodes. (See [1] and the references therein for an excellent overview of such approaches.) Unfortunately, in many applications, prior knowledge of the precise correlations in the data is unavailable, making it difficult or impossible to apply such distributed source coding techniques. This situation motivates collaborative, in-network processing and compression, in which unknown correlations and dependencies between the networked data can be learned and exploited by exchanging information between network nodes. However, the design and implementation of effective collaborative processing algorithms can be quite challenging, since they too rely on some prior knowledge of the anticipated correlations and depend on somewhat sophisticated communications and node processing capabilities.

This article describes a very different approach to the decentralized compression of networked data. Specifically, consider a compression of the form $y = Ax$, where $A = \{A_{i,j}\}$ is a $k \times n$ "sensing" matrix with far fewer rows than columns (i.e., $k \ll n$). The compressed data vector y is $k \times 1$, and therefore is much easier to store, transmit, and retrieve compared to the uncompressed networked data x. The theory of compressed sensing guarantees that, for certain matrices A, which are non-adaptive and often quite unstructured, x can be accurately recovered from y whenever x itself is compressible in some domain (e.g., frequency, wavelet, time) [2]–[5].

To carry the illustration further, and to motivate the approaches proposed in this article, let us look at a very concrete example. Suppose that *most* of the network nodes have the same nominal data value, but the few remaining nodes have different values. For instance, the values could correspond to security statistics or sensor readings at each node. The networked data vector in this case is mostly constant, except for a few deviations in certain locations. This minority may be of most interest in security or sensing applications. Clearly x is quite compressible; the nominal value plus the locations and values of the few deviant cases suffice for its specification.

Consider a few possible situations in this networked data compression problem. First, if the nominal value were known to all nodes, then the desired compression is accomplished simply by the deviant nodes sending that notification. Second, if the nominal value were not known, but the deviant cases were assumed to be isolated, then the nodes could simply compare their own values to those of their nearest neighbors to determine the nominal value and any deviation of their own. Again, notifications from the deviant nodes would provide the desired compression. There is a third, more general, scenario in which such simple local processing schemes can break down. Suppose that the nominal value is unknown to the nodes a priori, and that the deviant cases could be isolated or clustered. Since the deviant nodes may be clustered together, simply comparing values between neighboring nodes may not reveal them all, and perhaps not even the majority of them, depending on the extent of clustering. Indeed, distributed processing schemes in general are difficult to design without prior knowledge of the anticipated relations among data at neighboring nodes. This serves as a motivation for the theory and methods discussed here.

Compressed sensing offers an alternative measurement approach that does not require any specific prior signal knowledge and is an effective (and efficient) strategy in each of the situations described above. The values of all nodes can be recovered from the compressed data $y = Ax$, provided its size k is proportional to the number of deviant nodes. As we shall see, y can be efficiently computed in a distributed manner, and by virtue of its small size, it is naturally easy to store and transmit. In fact, in certain wireless network applications (see Wireless Sensor Networks in the Networked Data Compression in Action section of this article for details), y can be computed in the air itself, rather than in silicon! Thus, compressed sensing offers two highly desirable features for networked data analysis. The method is *decentralized*, meaning that distributed data can be encoded without a central controller, and *universal*, in the sense that sampling does not require a priori knowledge or assumptions about the data. For these reasons, the advantages of compressed sensing have already caught on in the research community, as evidenced by several recent works [6]–[10].

Compressed sensing basics

The theory of compressed sensing (CS) extends traditional sensing and sampling systems to a much broader class of signals. According to CS

theory, *any* sufficiently compressible signal can be accurately recovered from a small number of non-adaptive, randomized linear projection samples. For example, suppose that $x \in \mathbb{R}^n$ is m-sparse (i.e., it has no more than m nonzero entries) where m is much smaller than the signal length n. Sparse vectors are very compressible, since they can be completely described by the locations and amplitudes of the non-zero entries. Rather than sampling each element of x, CS directs us to first precondition the signal by operating on it with a diversifying matrix, yielding a signal whose entries are mixtures of the non-zero entries of the original signal. The resulting signal is then sampled k times to obtain a low-dimensional vector of observations. Overall, the acquisition process can be described by the observation model $y = Ax + \epsilon$, where the matrix A is a $k \times n$ CS matrix that describes the joint operations of preconditioning and subsampling, and ϵ represents errors due to noise or other perturbations. The main results of CS theory have established that if the number of CS samples is a small integer multiple greater than the number of non-zero entries in x, then these samples sufficiently "encode" the salient information in the sparse signal and an accurate reconstruction from y is possible. These results

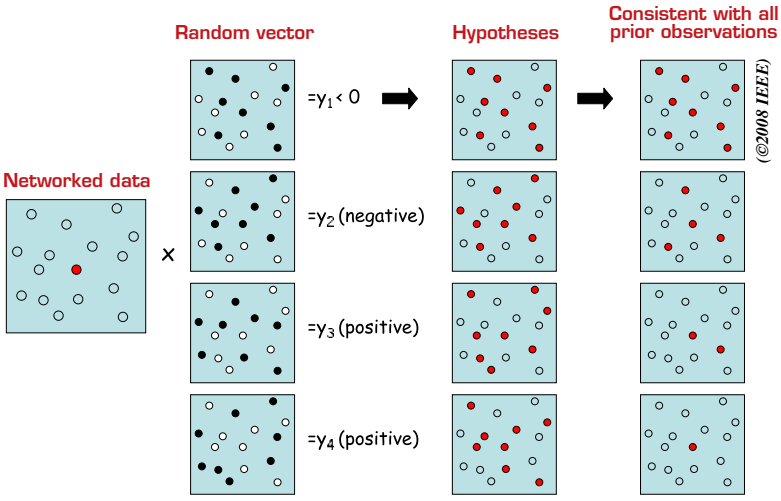


Figure 1: A simple reconstruction example on a network of $n = 16$ nodes. One distinguished sensor observes a positive value while the remaining $n - 1$ observe zero. The task is to identify which sensor is different by using as few observations as possible. In the CS approach, the data are projected onto random vectors, such as those depicted in the second column (where nodes indicated in black multiply their data value by −1 and those in white by +1). The third column shows that about $n/2$ hypothesis sensors are consistent with each random projection observation, but the number of hypotheses that are simultaneously consistent with *all* observations (shown in the fourth column) decreases exponentially with the number of observations. The random projection observations are approximately performing binary bisections of the hypothesis space, and only about log n observations are needed to determine which sensor reads the nonzero value.

are very promising because at least $2m$ pieces of information (the location and amplitude of each nonzero entry) are generally required to describe any m-sparse signal, and CS is an effective way to obtain this information in a simple, non-adaptive manner. The next few subsections explain, in some detail, how this is accomplished.

Compressed sensing for networked data

To illustrate the CS random projection encoding and reconstruction ideas, consider the simple reconstruction example (Figure 1). Suppose that in a network of n sensors, only one of the sensors is observing some positive value, while the rest of the sensors observe zero. The goal is to identify which sensor measures the nonzero value using a minimum number of observations. Consider making random projection observations of the data, where each observation is the projection of the sensor readings onto a random vector having entries ±1 each with probability 1/2. The value of each observation, along with knowledge of the random vector onto which the data was projected, can be used to identify a set of about $n/2$ hypothesis sensors that are consistent with that particular observation. The estimate of the anomalous sensor given k observations is simply the intersection of the hypothesis sets consistent with each of the k observations. It is easy to see that, on average, about log n observations are required before the correct (unique) sensor is identified. Define the ℓ_0 quasi-norm $\|z\|_0$ to be equal to the number of nonzero entries in the vector z. Then this simple procedure can be thought of as the solution of the optimization problem

$$\arg\min_z \|z\|_0 \text{ subject to } y = Az. \qquad (1)$$

Encoding requirements

Suppose that for some observation matrix A there is a nonzero m-sparse signal x such that the observations $y = Ax = 0$. Recovery of x is impossible in this setting, since the observations provide no information about the specific signal being observed. Matrices that are resilient to such ambiguities are those that satisfy the Restricted Isometry Property (RIP) [2], [11], [12]. Essentially, a $k \times n$ sensing matrix A with unit-normed rows (i.e., $\sum_{j=1}^{n} A_{i,j}^2 = 1$ for $i = 1,2,...,k$) is said to satisfy a RIP of order s whenever $\|Ax\|_2^2 \approx k\|x\|_2^2/n$ holds simultaneously for all s-sparse vectors $x \in \mathbb{R}^n$. The RIP is so-named because it describes matrices that impose

near-isometry (approximate length preservation) on a restricted set of subspaces (the subspaces of s-sparse vectors). In simpler terms, a matrix satisfies RIP if and only if vectors that are sufficiently sparse are not in the null space of the matrix.

In practice, sensing matrices that satisfy the RIP are easy to generate. It has been established that $k \times n$ matrices whose entries are independent and identically distributed realizations of certain zero-mean random variables with variance $1/n$ satisfy a RIP with very high probability when $k \geq const \cdot \log n \cdot m$ [2], [3], [13]. Physical limitations of real sensing systems motivate the unit-norm restriction on the rows of A, which essentially limits the amount of "sampling energy" allotted to each observation.

Decoding: Algorithms and bounds

Compressed sensing is a form of subsampling, so aliasing is present, and needs to be accounted for in the reconstruction process. The same compressed data could be generated by many n-dimensional vectors, but the RIP implies that only one of these is sparse. This might seem to require that any reconstruction algorithm must exhaustively search all sparse vectors, but fortunately the process is much more tractable. Given a vector of (noise-free) observations $y = Ax$, the unknown m-sparse signal x can be recovered exactly as the unique solution to

$$\arg\min_z \|z\|_1 \text{ subject to } y = Az, \tag{2}$$

where $\|z\|_1 = \sum^n{}_{i=1} |z_i|$ denotes the ℓ_1-norm, provided A satisfies RIP of order $2m$ [12]. The recovery procedure can be cast as a linear program, so solution methods are tractable even when n is very large.

Compressed sensing remains quite effective even when the samples are corrupted by additive noise, which is important from a practical point of view since any real system will be subjected to measurement inaccuracies. A variety of reconstruction methods have been proposed to recover (an approximation of) x when observations are corrupted by noise. For example, estimates $\hat{x}$ can be obtained as the solutions of either

$$\arg\min_z \|z\|_1 \text{ subject to } \|A^T(y-Az)\|_\infty \leq \lambda_1, \tag{3}$$

where $\|z\|_\infty = \max_{i=1,\ldots,n} |z_i|$ [5], or the penalized least squares minimization

$$\arg\min_z \left\{\|y-Az\|_2^2 + \lambda_2\|z\|_0\right\} \tag{4}$$

as proposed in [4], for appropriately chosen regularization constants λ_1 and λ_2 that each depend on the noise variance. In either case, the reconstruction error $\mathbb{E}\left[\|x-\hat{x}\|_2^2/n\right]$ decays at a rate of $(m \log n/k)$. In practice, the optimization (3) can be solved by a linear program, while (4) is often solved by convex relaxation—replacing the ℓ_0 penalty with the ℓ_1 penalty. The appeal of CS is readily apparent from the error rate which (ignoring the logarithmic factor) is proportional to m/k, the variance of an estimator of m parameters from k observations. In other words, CS is able to both identify the locations and estimate the amplitudes of the non-zero entries without any specific prior knowledge about the signal except the assumption of sparsity.

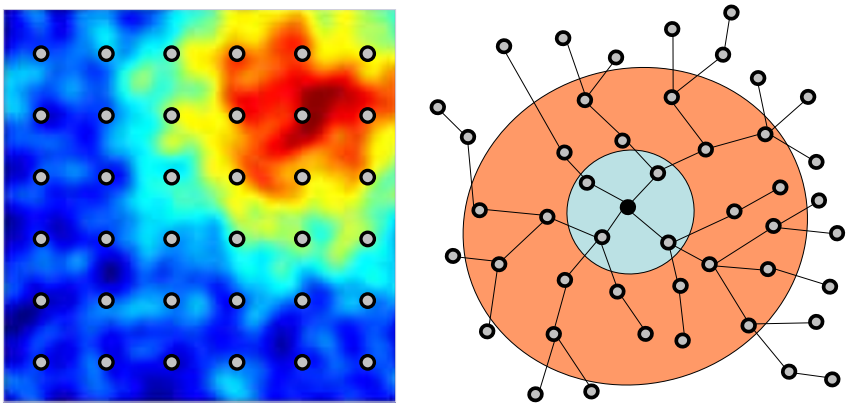
Transform domain sparsity

Suppose the observed signal x is not sparse, but instead a suitably transformed version is. Specifically, let T be a transformation matrix, and assume that $\theta = Tx$ is sparse. The CS observations can be written as $y = Ax = AT^{-1}\theta$. If A is a random CS matrix satisfying the RIP, then in many cases so is the product matrix AT^{-1} [13]. Consequently, the CS observation process does not require prior knowledge of the domain in which the data are compressible. The sparse vector θ (and hence x) can be accurately recovered from y using the reconstruction techniques described above. For example, in the noiseless setting one can solve

$$\hat{\theta} = \arg\min_z \|z\|_1 \text{ subject to } y = AT^{-1}z, \tag{5}$$

to obtain an exact reconstruction of the transform coefficients of x. Note that, while the samples do not require selection of an appropriate sparsifying transform, the reconstruction does.

Often, signals of interest will not be exactly sparse, but instead most of the energy is concentrated on a relatively small set of entries while the remaining entries are very small. The degree of effective sparsity of such signals can be quantified with respect to a given basis. Formally, for a signal x let x^s be the approximation of x formed by retaining the s coefficients having largest magnitude in the transformed representation $\theta = Tx$. Then x is called *α-compressible* if the approximation error obeys



(a) (b)

Figure 2: Sparsifying transformation techniques depend on network topologies. The smoothly varying field in (a) is monitored by a network of wireless sensors deployed uniformly over the region, and standard transform techniques can be used to sparsify the networked data. For more abstract topologies, graph wavelets can be effective. In (b), the graph (Haar) wavelet coefficient at the location of the black node and scale three is given by the difference of the average data values at the nodes in the red and blue regions.

$$\frac{\|x-x^s\|_2^2}{n} \le const \cdot s^{-2\alpha} \qquad (6)$$

for some $\alpha = \alpha(x,T) > 0$. This model describes, for example, signals whose ordered (transformed) coefficient amplitudes exhibit power-law decay. Such behavior is associated with images that are smooth or have bounded variation [3], [11], and is often observed in the wavelet coefficients of natural images. In this setting, CS reconstruction techniques can again be applied to obtain an estimate of the transformed coefficients directly. For example, the solutions of optimizations analogous to (3) and (4) yield estimates whose estimation error decays at the rate $(\log n/k)^{2\alpha/2\alpha+1}$, quantifying the simultaneous balancing of the errors due to approximation and estimation [4]. This result guarantees that even when signals are only approximately sparse, consistent estimation is still possible.

Sparsifying networked data

Compressed sensing can be very effective when x is sparse or highly compressible in a certain basis or dictionary. But, while transform-based compression is well-developed in traditional signal and image processing domains, the understanding of sparsifying/compressing bases for networked data is far from complete. There are, however, a few promising new approaches to the design of transforms for networked data, some of which are described below.

Spatial compression

Suppose a wireless sensor network is deployed to monitor a certain spatially-varying phenomenon, such as temperature, light, or moisture. The physical field being measured can be viewed as a signal or image with a degree of spatial correlation or smoothness. If the sensors are geographically placed in a uniform fashion, such as in Figure 2(a), then sparsifying transforms may be readily borrowed from traditional signal processing. In these settings, the sensor locations can be viewed as *sampling locations* and tools like the discrete Fourier transform (DFT) or discrete wavelet transform (DWT) may be used to sparsify the networked data. In more general settings, wavelet techniques can be extended to also handle nonuniform distribution of sensors [14].

Graph wavelets

Standard signal transforms cannot be applied in more general situations. For example, many network monitoring applications rely on the analysis of traffic levels at the network nodes. Changes in the behavior of traffic levels can be indicative of variations in network usage, component failures, or malicious activities. There are strong correlations between traffic levels at different nodes, but the topology and routing affect the nature of these relationships in complex ways. Graph wavelets, developed with these challenges in mind, adapt the design principles of the DWT to arbitrary networked data [15].

To understand graph wavelets, it is useful to first consider the Haar wavelet transform, which is the simplest form of DWT. The Haar wavelet coefficients are essentially obtained as digital differences of the data at different scales of aggregation. The coefficients at the first scale are differences between neighboring data points, and those at subsequent spatial scales are computed by first aggregating data in neighborhoods (dyadic intervals in one dimension and square regions in two dimensions) and then computing differences between neighboring aggregations.

Graph wavelets are a generalization of this construction, where the number of hops between nodes in a network provides a natural distance

measure that can be used to define neighborhoods. The size of each neighborhood (with radius defined by the number of hops) provides a natural measure of scale, with smaller sizes corresponding to finer spatial analysis of the networked data. Graph wavelet coefficients are then defined by aggregating data at different scales, and computing differences between aggregated data, as shown in Figure 2(b). Further details and generalizations of this can be found in [15].

Diffusion wavelets

Diffusion wavelets provide an alternative approach to constructing a multi-scale representation for networked data. Unlike graph wavelets, which produce an overcomplete dictionary, diffusion wavelets produce an orthonormal basis tailored to a specific network by analyzing eigenvectors of a diffusion matrix derived from the network adjacency matrix (hence the name "diffusion wavelets"). The resulting basis vectors are generally localized to neighborhoods of varying size and may also lead to a sparsifying representation of networked data. A thorough treatment of this topic can be found in [16].

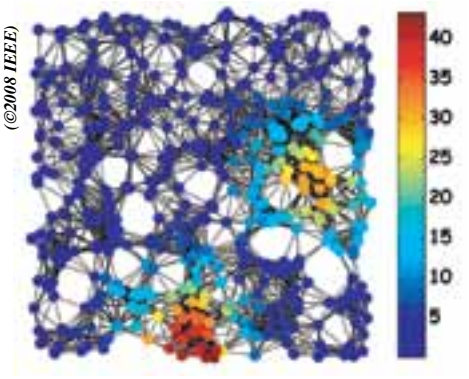
One example of sparsification using diffusion wavelets is shown in Figure 3, where the node data correspond to traffic rates through routers in a computer network. There are several highly localized regions of activity, while most of the remaining network exhibits only moderate levels of traffic. The traffic data are sparsely represented in the diffusion wavelet basis, and a small number of coefficients can provide an accurate estimate of the actual traffic patterns.

Networked data compression in action

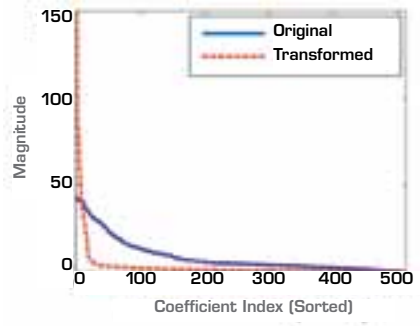
This section describes two techniques for obtaining projections of networked data onto general vectors, which can be thought of as the rows of the sensing matrix A. The first approach described below assumes that the network is any general multihop network. This model could explain, for example, wireless sensor networks, wired local area networks, or even portions of the Internet. In the multihop setting, the projections can be computed and delivered to every subset of nodes in the network using gossip/consensus techniques, or they might be delivered to a single point using clustering and aggregation. The second, more specific, approach described below is motivated by many wireless sensor networking applications in which explicit routing information is difficult to obtain and maintain in real time. In this setting, sensors contribute their measurements in a joint fashion by simultaneous wireless transmissions to a distant processing location, and the observations are accumulated and processed at that (single) destination point.

Compressed sensing for networked data storage and retrieval

In general multihop networks, two simple steps can be used for the decentralized computation and distribution of each CS observation of the form $y_i = \sum_{j=1}^{n} A_{i,j} x_j$, $i = 1,...,k$:



(a) Local traffic rates



(b) Ordered Coefficients

Figure 3: An illustration of the compressibility of spatially correlated networked data using diffusion wavelets. The actual networked data shown in (a) are not sparse, but can be represented with a small number of diffusion wavelet coefficients, as seen in (b).

Step 1: Each of the n sensors, $j = 1,...,n$, locally computes the term $A_{i,j} x_j$ by multiplying its data with the corresponding element of the sensing matrix. The sensing matrix can be generated in a distributed fashion by letting each node locally generate a realization of $A_{i,j}$ using a pseudo-random number generator seeded with its identifier. (In this example, the integers $j = 1,...,n$ serve as the identifiers.) Given the identifiers of the nodes, the destination node(s) can also easily generate the random vectors $\{A_{i,j}\}_{i=1}^{k}$ for each sensor $j = 1,...,n$.

Step 2: The local terms $A_{i,j} x_j$ are simultaneously aggregated and distributed across the network using randomized gossip, which is a simple iterative decentralized algorithm for computing linear functions such as $y_i = \sum_{j=1}^{n} A_{i,j} x_j$ (see Figure 4). Note that gossip algorithms are highly resilient to node failures because: (i) each node only exchanges information with its immediate neighbors, and (ii) when they terminate, the value of y_i is available at every node in the network.

Since the above procedure ensures that the networked data projections are known at every node, a user can query any node in the network and compute $\hat{x}$ via one of the reconstruction methods

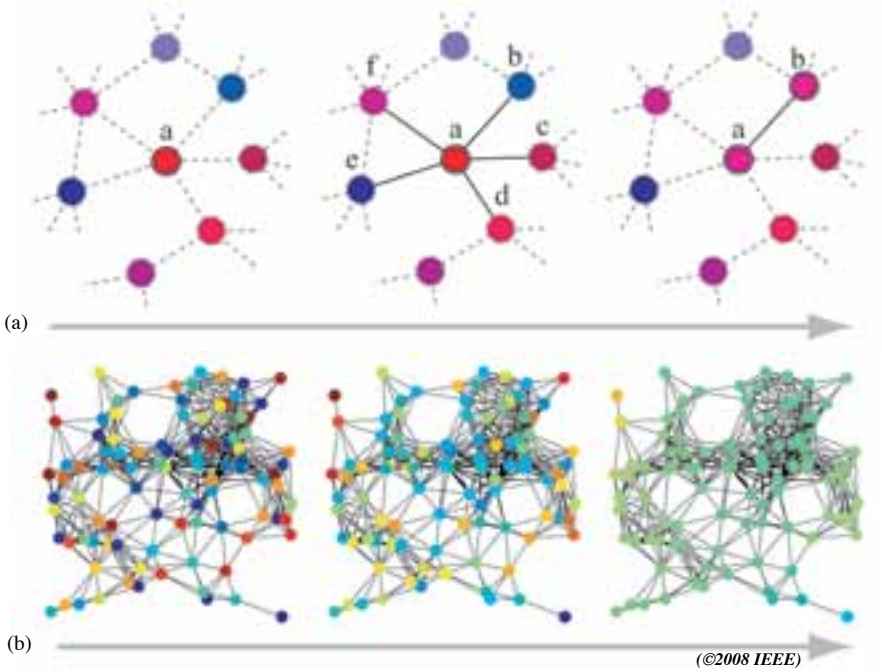


Figure 4: **Randomized gossip:** (a) depicts one iteration, in which the color of a node corresponds to its local value. To begin, the network is initialized to a state where each node has a value $x_i(0)$, $i = 1,...,n$. Then in an iterative, asynchronous fashion, a random node a "activates" and chooses one of its neighbors b at random. The two nodes then "gossip" by exchanging their values $x_a(t)$ and $x_b(t)$, or in the CS setting the values multiplied by pseudo-random numbers, and perform the update $x_a(t + 1) = x_b(t + 1) \leftarrow (x_a(t) + x_b(t))/2$, while the data at all the other nodes remains unchanged. In (b), we have an example network of 100 nodes with: (i) random initial values (left), (ii) after each node has communicated five times with each of its neighbors (middle), and (iii) after each node has communicated 50 times with each of its neighbors (right). It can be shown that for this simple procedure, $x_i(t)$ converges to the average of the initial values, $1/n \sum_{j=1}^{n} x_j(0)$, at every node in the network as t tends to infinity.

outlined in the Compressed Sensing Basics section. Further, this can be quite an efficient procedure in many scenarios. For example, in networks with power-law degree distributions such as the Internet, an optimized gossip algorithm uses on the order of kn transmissions to compute k CS observations [17], generally $k \ll n$. So this is much more efficient than exhaustively exchanging raw data values, which would take about n^2 transmissions. Of course, if the location of the node to be queried is fixed a priori—and if the network provides reliable routing service—then it may be more efficient to replace gossip computation with aggregation up a spanning tree or around a cycle. For more on using gossip algorithms to compute/distribute compressed data in multihop networks, see [7].

Compressed sensing in wireless sensor networks

A typical wireless sensor network, as in Figure 5, consists of a large number of small, inexpensive wireless sensors, spatially distributed over a region of interest that can sense the physical environment in a variety of modalities. The essential task in many applications of sensor networks is to extract some relevant information from distributed data and then wirelessly deliver it to a distant destination, called the fusion center (FC). While this task can be accomplished in a number of ways, one particularly attractive technique corresponds to delivering random projections of the sensor networked data to the FC by exploiting recent results on uncoded (analog) coherent transmission schemes in wireless sensor networks [18]–[21]. The proposed distributed communication architecture—introduced in [6], [8], and refined in [22]—requires only one (network) transmission per random projection and is based on the notion of so-called "matched source-channel communication" [20], [21]. Here, the CS projection observations are simultaneously calculated (by the superposition of radio waves) and communicated using amplitude-modulated coherent transmissions of randomly weighted sensed values directly from the sensor nodes to the FC via the air interface. Algorithmically, sensor nodes sequentially perform the following steps in order to communicate k random projections of the networked data to the FC:

Step 1: Each of the n sensors locally draws k elements of the random projection vectors $\{A_{i,j}\}_{i=1}^{k}$ by using its network address as the seed of a pseudo-random number generator. Given the network addresses of the nodes, the FC can also easily reconstruct the random vectors $\{A_{i,j}\}_{i,j=1}^{k,n}$.

Step 2: The sensor at location j multiplies its measurement x_j with $\{A_{i,j}\}_{i=1}^{k}$ to obtain a k-tuple

$$v_j = (A_{1,j}\, x_j,...,A_{k,j}\, x_j)^T,\ j = 1,...,n, \tag{7}$$

and all the nodes coherently transmit their respective v_j's in an analog fashion over the network-to-FC air interface using k transmissions. Because of the additive nature of radio waves, the corresponding received signal at the FC at the end of the k-th transmission is given by

$$y = \sum_{j=1}^{n} v_j + \epsilon = Ax + \epsilon, \quad (8)$$

where ϵ is the noise generated by the communication receiver circuitry of the FC. The steps above correspond to a completely decentralized way of delivering k random projections of the networked data to the FC by employing k (network) transmissions. The final estimate $\hat{x}$ can be computed at the FC via any of the methods outlined earlier. As noted earlier, the main advantage of using this approach for computing random projections is that it can be implemented *without* any complex routing information and as a result might be a more suitable and scalable option in many sensor networking applications.

Conclusions and extensions

This article has described how compressed sensing techniques could be utilized to reconstruct sparse or compressible networked data in a variety of practical settings, including general multihop networks and wireless sensor networks. Compressed sensing provides two key features, universal sampling and decentralized encoding, making it a promising new paradigm for networked data analysis. The focus here was primarily on managing resources during the encoding process, but it is important to note that the decoding step also poses a significant challenge. Indeed, the study of efficient decoding algorithms remains at the forefront of current research [23]–[25].

In addition, specialized measurement matrices, such as those resulting from Toeplitz-structured matrices [26] and the incoherent basis sampling methods described in [27], lead to significant reductions in the complexity of convex decoding methods. Fortunately, the sampling matrices inherent to these methods can be easily implemented using the network projection approaches described above. For example, Toeplitz-structured CS matrices naturally result when each node uses the same random number generation scheme and seed value, in which each node advances its own random sequence by its unique (integer) identifier at initialization. Similarly, random samples from any orthonormal basis (the observation model described in [27]) can easily be obtained in the settings described above if each node is preloaded with its weights for each basis element in the corresponding orthonormal transformation matrix. For each observation, the requesting node (or fusion center) broadcasts a random integer between 1 and n to the nodes to specify which transform coefficient from the predetermined basis should be obtained, and the projection is delivered using any suitable method described above.

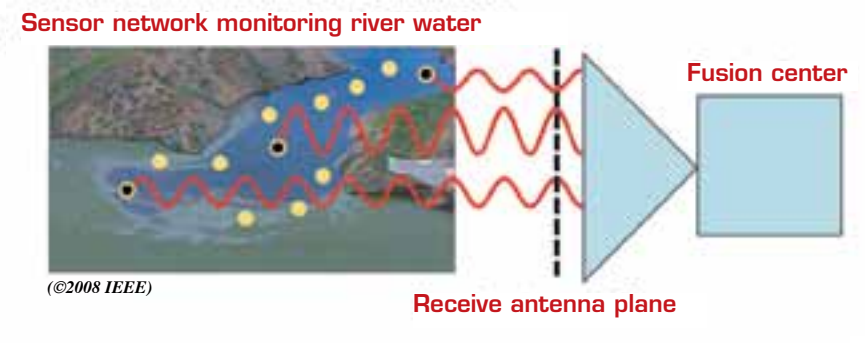


Figure 5: An illustration of a wireless sensor network and fusion center. A number of sensor nodes monitor the river water for various forms of contamination and periodically report their findings over the air to the fusion center. CS projection observations are obtained by each sensor transmitting a sinusoid with amplitude given by the product of the sensor measurement and a pseudo-random weight. When the transmissions arrive in phase at the fusion center, the amplitude of the resulting received waveform is the sum of the component wave amplitudes.

Finally, it is worth noting that matrices satisfying the RIP also approximately preserve additional geometrical structure on subspaces of sparse vectors, such as angles and inner products, as shown in [28]. A useful consequence of this result is that an ensemble of CS observations can be "data mined" for events of interest [29], [30]. For example, consider a network whose data may contain an anomaly that originated at one of m candidate nodes. An ensemble of CS observations of the networked data, collected without any a priori information about the anomaly, can be analyzed "post-mortem" to accurately determine which candidate node was the likely source of the anomaly. Such extensions of CS theory suggest efficient and scalable techniques for monitoring large-scale distributed networks, many of which can be performed without the computational burden of reconstructing the complete networked data.

About the authors

Robert Nowak received the BS (with highest distinction), MS, and PhD degrees in electrical engineering from the University of Wisconsin-Madison in 1990, 1992, and 1995, respectively. He was a Postdoctoral Fellow at Rice University in 1995–1996, an Assistant Professor at Michigan

State University from 1996-1999, held Assistant and Associate Professor positions at Rice University from 1999-2003, and was a Visiting Professor at INRIA in 2001. Nowak is now the McFarland-Bascom Professor of Engineering at the University of Wisconsin-Madison. He has served as an Associate Editor for the IEEE Transactions on Image Processing, the Secretary of the SIAM Activity Group on Imaging Science, and is currently an Associate Editor for the ACM Transactions on Sensor Networks. He was General Chair for the 2007 IEEE Statistical Signal Processing workshop and Technical Program Chair for the 2003 IEEE Statistical Signal Processing Workshop and the 2004 IEEE/ACM International Symposium on Information Processing in Sensor Networks. Dr. Nowak received the General Electric Genius of Invention Award in 1993, the National Science Foundation CAREER Award in 1997, the Army Research Office Young Investigator Program Award in 1999, the Office of Naval Research Young Investigator Program Award in 2000, and IEEE Signal Processing Society Young Author Best Paper Award in 2000. He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE). His research interests include signal processing, machine learning, imaging and network science, and applications in communications, bioimaging, and systems biology.

Michael Rabbat earned the BSc from the University of Illinois at Urbana-Champaign (2001), the MSc degree from Rice University (2003), and the PhD from the University of Wisconsin–Madison (2006), all in electrical engineering. He is currently an Assistant Professor at McGill University. He was a visiting researcher at Applied Signal Technology, Inc. during the summer of 2003. He received the Best Student Paper award at the 2004 ACM/IEEE Conference on Information Processing in Sensor Networks, Outstanding Student Paper Honorable Mention at the 2006 Conference on Neural Information Processing Systems, and the Harold A. Peterson Thesis Prize. His current research focuses on distributed information processing in sensor networks, network monitoring, and network inference.

Jarvis Haupt (Postdoctoral Research Associate, Rice University) received his PhD in Electrical Engineering from the University of Wisconsin–Madison in 2009. His primary research interests include sensing, processing, and recovery techniques for high-dimensional data, and in particular, sparsity-aware methods such as compressive sensing and adaptive sensing. His research establishes the significant efficiency and performance improvements for these techniques, relative to conventional sampling and recovery methods, in a variety of application domains including imaging and remote sensing, networks, and analog and digital communications.

Waheed U. Bajwa received the BE (with Honors) degree in electrical engineering from the National University of Sciences and Technology, Islamabad, Pakistan in 2001, and the MS and PhD degrees in electrical engineering from the University of Wisconsin–Madison, Madison in 2005 and 2009, respectively. He is currently a Postdoctoral Research Associate in the Program in Applied and Computational Mathematics at Princeton University. In addition, he has more than three years of industry experience, including a summer position at GE Global Research, Niskayuna, NY in 2006. His research interests include high-dimensional inference, statistical signal processing, wireless communications, learning theory, and applications in biological sciences, cyber-physical systems, radar and image processing, and cognitive/ad-hoc networks.

References

[1] S. S. Pradhan, J. Kusuma, and K. Ramchandran. Distributed compression in a dense microsensor network. *IEEE Signal Processing Mag*., 19(2):51–60, March 2002.

[2] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Trans. Inform. Theory*, 51(12):4203–4215, December 2005.

[3] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, April 2006.

[4] J. Haupt and R. Nowak. Signal reconstruction from noisy random projections. *IEEE Trans. Inform. Theory*, 52(9):4036–4048, September 2006.

[5] E. Candès and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313-2351, December 2007.

[6] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. Compressive wireless sensing. In *Proc. IPSN'06*, pages 134–142, Nashville, TN, April 2006.

[7] M. Rabbat, J. Haupt, A. Singh, and R. Nowak. Decentralized compression and predistribution via randomized gossiping. In *Proc. IPSN'06*, pages 51–59, Nashville, TN, April 2006.

[8] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. A universal matched source-channel communication scheme for wireless sensor ensembles. In *Proc. ICASSP'06*, pages 1153–1156, Toulouse, France, May 2006.

[9] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk. Distributed compressed sensing. pre-print. [Online]. Available: http://www.ece.rice.edu/~drorb/pdf/DCS112005.pdf

[10] W. Wang, M. Garofalakis, and K. Ramchandran. Distributed sparse random projections for refinable approximation. In *Proc. IPSN'07*, pages 331–339, Cambridge, MA, April 2007.

[11] E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. Inform. Theory*, 52(12):5406–5425, December 2006.

[12] E. J. Candès. The restricted isometry property and its implications for compressed sensing. In *C. R. Acad. Sci., Ser. I*, vol. 346, pages 589–592, Paris, 2008.

[13] R. Baraniuk, M. Davenport, R. A. DeVore, and M. B. Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253-263, December 2008.

[14] R. Wagner, R. Baraniuk, S. Du, D. Johnson, and A. Cohen. An architecture for distributed wavelet analysis and processing in sensor networks. In *Proc. IPSN'06*, pages 243–250, Nashville, TN, April 2006.

[15] M. Crovella and E. Kolaczyk. Graph wavelets for spatial traffic analysis. In Proc. *IEEE Infocom.*, vol. 3, pages 1848–1857, March 2003.

[16] R. Coifman and M. Maggioni. Diffusion wavelets. *Applied Computational and Harmonic Analysis*, 21(1):53–94, July 2006.

[17] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Inform. Theory*, 52(6):2508–2530, June 2006.

[18] M. Gastpar and M. Vetterli. Source-channel communication in sensor networks. In *Proc. IPSN'03*, pages 162–177, Palo Alto, CA, April 2003.

[19] K. Liu and A. M. Sayeed. Optimal distributed detection strategies for wireless sensor networks. In *Proc. 42nd Annual Allerton Conference on Commun., Control and Comp.*, October 2004.

[20] M. Gastpar and M. Vetterli. Power, spatio-temporal bandwidth, and distortion in large sensor networks. *IEEE Journal Select. Areas Commun.*, 23(4):745–754, April 2005.

[21] W. U. Bajwa, A. M. Sayeed, and R. Nowak. Matched source-channel communication for field estimation in wireless sensor networks. In *Proc. IPSN'05*, pages 332–339, Los Angeles, CA, April 2005.

[22] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak. Joint source-channel communication for distributed estimation in sensor networks. *IEEE Trans. Inform. Theory*, 53(10):3629–3653, October 2007.

[23] A. C. Gilbert and J. Tropp. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform.* Theory, 53(12):4655-4666, December 2007.

[24] M. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal Select. Topics in Signal Processing*, 1(4):586-597, 2007.

[25] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior point method for large-scale l1-regularized least squares. *IEEE Journal Select. Topics in Signal Processing*, 1(4):606-617, 2007.

[26] W. Bajwa, J. Haupt, G. Raz, and R. Nowak. Toeplitz-structured compressed sensing matrices. In *Proc. SSP'07*, pages 294–298, Madison, WI, August 2007.

[27] E. Candès and J. Romberg. Sparsity and incoherence in compressive sampling. *Inverse Problems*, 23(3):969–985, 2006.

[28] J. Haupt and R. Nowak. *A generalized restricted isometry property*. University of Wisconsin - Madison, Tech. Rep. ECE-07-1, May 2007.

[29] J. Haupt and R. Nowak. Compressive sampling for signal detection. In *Proc. ICASSP'07*, Honolulu, HI, April 2007.

[30] J. Haupt, R. Castro, R. Nowak, G. Fudge, and A. Yeh. Compressive sampling for signal classification. In *Proc. 40th Asilomar Conference on Signal, Systems, and Computers*, pages 1430–1434, Pacific Grove, CA, October 2006.