Bayesian Compressive Sensing
via Belief Propagation

Dror Baron,[1] Shriram Sarvotham,[2] and Richard G. Baraniuk[3] [*]

[1]Department of Electrical Engineering, Technion – Israel Institute of Technology; Haifa, Israel

[2]Halliburton; Houston, TX

[3]Department of Electrical and Computer Engineering, Rice University; Houston, TX

December 25, 2008

Abstract

Compressive sensing (CS) is an emerging field based on the revelation that a small collection of linear projections of a sparse signal contains enough information for stable, sub-Nyquist signal acquisition. When a statistical characterization of the signal is available, Bayesian inference can complement conventional CS methods based on linear programming or greedy algorithms. We perform approximate Bayesian inference using belief propagation (BP) decoding, which represents the CS encoding matrix as a graphical model. Fast encoding and decoding is provided using sparse encoding matrices, which also improve BP convergence by reducing the presence of loops in the graph. To decode a length-N signal containing K large coefficients, our CS-BP decoding algorithm uses $O(K \log(N))$ measurements and $O(N \log^2(N))$ computation. Finally, sparse encoding matrices and the CS-BP decoding algorithm can be modified to support a variety of signal models and measurement noise.

1 Introduction

Many signal processing applications require the identification and estimation of a few significant coefficients from a high-dimensional vector. The wisdom behind this is the ubiquitous compressibility of signals: in an appropriate basis, most of the information contained in a signal often resides in just a few large coefficients. Traditional sensing and processing first acquires the entire data, only to later throw away most coefficients and retain the few significant ones [2]. The ground-breaking work in *compressive sensing* (CS), pioneered by Candés et al. [3] and Donoho [4], demonstrates that the information contained in the few large coefficients can be captured (encoded) by a small number of random linear projections. The signal can then be decoded from these random projections.

1.1 Compressive sensing

Sparsity and random encoding: In a typical compressive sensing (CS) setup, a signal vector $x \in \mathbb{R}^N$ has the form $x = \Psi\theta$, where $\Psi \in \mathbb{R}^{N \times N}$ is an orthonormal basis, and $\theta \in \mathbb{R}^N$ satisfies $\|\theta\|_0 = K \ll N$.[1] Owing to the *sparsity* of x relative to the basis Ψ, there is no need to sample all N values of x. Instead, the CS theory establishes that x can be decoded from a small number of projections onto an incoherent set of measurement vectors [3, 4]. To measure (encode) x, we compute $M \ll N$ linear projections of x via the matrix-vector multiplication, $y = \Phi x$, where $\Phi \in \mathbb{R}^{M \times N}$ is the encoding matrix.

In addition to *strictly sparse* signals where $\|\theta\|_0 = K$, other *signal models* are possible. *Approximately sparse* signals have $K \ll N$ large coefficients, while remaining coefficients are small but not necessarily zero; *compressible* signals have coefficients that, when sorted, obey a power law. Similarly, both *noiseless* and *noisy* measurements may be considered. We emphasize noiseless measurement of approximately sparse signals in the paper.

Decoding via sparsity: Our goal is to decode x given y and Φ. Although decoding x from $y = \Phi x$ appears to be an ill-posed inverse problem, the prior knowledge of sparsity in x enables to decode x from $M \ll N$ measurements. Decoding relies on an optimization, which searches for the sparsest coefficients θ that agree with the measurements y. If M is sufficiently large and θ is strictly sparse, then θ is the solution to the ℓ_0 minimization,

$$\widehat{\theta} = \arg\min \|\theta\|_0 \quad \text{s.t.} \ \ y = \Phi\Psi\theta.$$

Unfortunately, solving this ℓ_0 optimization is NP-complete [5].

The revelation that supports the CS theory is that a computationally tractable optimization problem yields an equivalent solution. We need only solve for the ℓ_1-sparsest coefficients that agree with the measurements y [3, 4],

$$\widehat{\theta} = \arg\min \|\theta\|_1 \quad \text{s.t.} \ \ y = \Phi\Psi\theta, \tag{1}$$

as long as $\Phi\Psi$ satisfies some technical conditions, which are satisfied with overwhelming probability when entries of Φ are independent and identically distributed (iid) Gaussian random variables [3]. This ℓ_1 optimization problem (1), also known as *Basis Pursuit* [6], can be solved with linear programming methods. The ℓ_1 decoder requires only $M = O(K \log(N/K))$ projections [7, 8]. However, encoding by a fully Gaussian Φ is slow, and ℓ_1 decoding requires at least quadratic computation in general [4, 9].

1.2 Fast CS decoding

While ℓ_1 decoders figure prominently in the CS literature, their quadratic complexity still renders them impractical for many applications. For example, current digital cameras acquire images with $N = 10^6$ pixels or more, and fast decoding is critical. The slowness of ℓ_1 decoding has motivated a proliferation of research into faster algorithms.

Iterative greedy algorithms have been developed for decoding. The *Orthogonal Matching Pursuit* (OMP) [10] algorithm, for example, iteratively selects the vectors from the matrix

---

[1]We use $\|\cdot\|_0$ to denote the ℓ_0 "norm" that counts the number of non-zero elements.

$\Phi\Psi$ that contain most of the energy of the measurement vector y. OMP has been proven to successfully decode the acquired signal with high probability [10, 11]. Algorithms inspired by OMP, such as tree matching pursuit [12], stagewise OMP [13], CoSaMP [14], and Subspace Pursuit [15] have been shown to attain similar guarantees to those of their optimization-based counterparts [16–18].

While the CS algorithms discussed above typically use a dense Φ matrix, a class of methods has emerged that employ Φ that is itself sparse. Cormode and Muthukrishnan proposed fast algorithms based on group testing [19, 20], which considers subsets of signal coefficients in which we expect at most one "heavy hitter" coefficient to lie. Gilbert et al. [21] propose the Chaining Pursuit algorithm, which works best for extremely sparse signals.

1.3   Bayesian CS

CS decoding algorithms implicitly rely on the sparsity of the signal x. In some applications, a statistical characterization of the signal is available, and Bayesian inference offers the potential for more precise estimation of x or a reduction in the number of CS measurements. Ji et al. [22] have proposed a Bayesian CS framework where relevance vector machines are used for signal estimation. For certain types of hierarchical priors, their method can approximate the posterior density of x and is somewhat faster than ℓ_1 decoding. Seeger and Nickisch [23] extend these ideas to experimental design, where the encoding matrix is designed sequentially based on previous measurements.

1.4   Contributions

In this paper, we use a sparse Φ matrix and *belief propagation* (BP) [24–30] to accelerate CS encoding and decoding under the Bayesian framework. We call our algorithm CS-BP. Although we emphasize a two-state mixture Gaussian model as a prior for sparse signals, CS-BP is flexible to variations in the signal and measurement models.

Sparse CS matrix: The fully Gaussian CS encoding matrices [3, 4] are reminiscent of Shannon's random code constructions. However, although dense matrices capture the information content of sparse signals, they may not be amenable to fast encoding and decoding. *Low density parity check* (LDPC) codes [31, 32] offer an important insight: encoding and decoding are fast, because multiplication by a sparse matrix is fast; nonetheless, LDPC codes achieve rates close to the Shannon limit. Indeed, in a previous paper [33], we used an LDPC-like sparse Φ for the special case of noiseless measurement of strictly sparse signals; similar matrices were also proposed for CS by Berinde and Indyk [34].

Encoder: We encode (measure) the signal using sparse Bernoulli ($\{0, 1\}$) or Rademacher ($\{0, 1, -1\}$) LDPC-like Φ matrices. Because entries of Φ are restricted to $\{0, 1, -1\}$, encoding only requires sums and differences of small subsets of coefficient values of x. The design of Φ, including characteristics such as column and row weights, is based on the relevant signal and measurement models, as well as the accompanying decoding algorithm.

Decoder: We represent the sparse Φ as a sparse bipartite graph. In addition to accelerating the algorithm, the sparse structure reduces the number of loops in the graph, and thus assists the convergence of a message passing method that solves a Bayesian inference

problem. Our estimate for x explains the measurements while offering the best match to the prior. We employ BP in a manner similar to LDPC channel decoding [29, 31, 32]. To decode a length-N signal containing K large coefficients, our CS-BP decoding algorithm uses $M = O(K \log(N))$ measurements and $O(N \log^2(N))$ computation.

The remainder of the paper is organized as follows. Section 2 defines our signal model and Section 3 describes our sparse CS-LDPC encoding matrix. Exact Bayesian inference appears in Section 4; approximate inference is provided by the CS-BP decoding process, which is described in Section 5. The performance of CS-BP is demonstrated numerically in Section 6. Variations and applications are discussed in Section 7, and Section 8 concludes. Technical details are included in appendices.

2   Mixture Gaussian signal model

We focus on a two-state mixture Gaussian model [35–37] as a prior that succinctly captures our prior knowledge about approximate sparsity of the signal. This model was originally proposed for a CS setting in our technical report [1], and also used by He and Carin [38]. More formally, let $X = [X(1), \ldots, X(N)]$ be a random vector in $\mathbb{R}^N$, and consider the signal $x = [x(1), \ldots, x(N)]$ as an outcome of X. Because our approximately sparse signal consists of a small number of large coefficients while remaining coefficients are small, we associate each *probability density function* (pdf) $f(X(i))$ with a state variable $Q(i)$ that can take on two values. Large and small magnitudes correspond to zero mean Gaussian distributions with high and low variances, which are implied by $Q(i) = 1$ and $Q(i) = 0$, respectively,

$$f(X(i)|Q(i) = 1) \sim \mathcal{N}(0, \sigma_1^2), \quad \text{and} \quad f(X(i)|Q(i) = 0) \sim \mathcal{N}(0, \sigma_0^2),$$

with $\sigma_1^2 > \sigma_0^2$. Let $Q = [Q(1), \ldots, Q(N)]$ be the state random vector associated with the signal; the actual configuration $q = [q(1), \ldots, q(N)] \in \{0, 1\}^N$ is one of 2^N possible outcomes. We assume that $Q(i)$'s are iid.[2] To ensure that we have approximately K large coefficients, we choose the probability mass function of the state variable $Q(i)$ to be Bernoulli with $\Pr(Q(i) = 1) = S$ and $\Pr(Q(i) = 0) = 1 - S$, where $S = K/N$ is the *sparsity rate.*

The resulting model for signal coefficients is a two-state mixture Gaussian distribution. This mixture model is completely characterized by three parameters: the sparsity rate S and the variances σ_0^2 and σ_1^2 of the Gaussian pdf's corresponding to each state. Figure 1 illustrates the two-state mixture Gaussian signal model.

Mixture Gaussian models have been successfully employed in image processing and inference problems, because they are simple yet effective in modeling real-world signals [35–37]. Theoretical connections have also been made between wavelet coefficient mixture models and the fundamental parameters of Besov spaces, which have proved invaluable for characterizing real-world images. Moreover, arbitrary densities with a finite number of discontinuities can be approximated arbitrarily closely by increasing the number of states and allowing non-zero means [39]. We leave these extensions for future work, and focus on two-state mixture Gaussian distributions for modeling the signal coefficients.

---

[2]The model can be extended to capture dependencies between coefficients, as suggested by Ji et al. [22].

$$\Pr(Q=0) \qquad \Pr(Q=1)$$
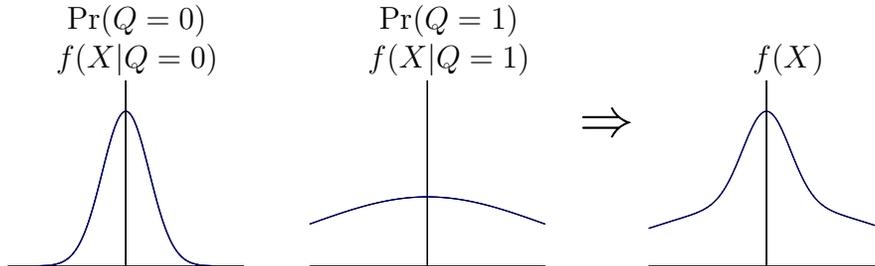$$f(X|Q=0) \qquad f(X|Q=1) \qquad\qquad f(X)$$

Figure 1: *Mixture Gaussian model for signal coefficients. The distribution of X conditioned on the two state variables, $Q = 0$ and $Q = 1$, is depicted. Also shown is the overall distribution for X.*

3   Sparse encoding

We use a sparse Φ matrix to accelerate CS encoding and decoding. Our CS encoding matrices are dominated by zero entries, with a small number of non-zeros in each row and each column. We focus on *CS-LDPC matrices* whose nonzero entries are $\{-1, 1\}$;[3] each measurement involves only sums and differences of a small subset of coefficients of x. A CS-LDPC Φ can be represented as a bipartite graph G, which is also sparse. Each edge of G connects a coefficient node $x(i)$ to an encoding node $y(j)$, and corresponds to a non-zero entry of Φ.

In addition to the core structure of Φ, we may introduce other constraints to tailor the measurement process to the signal model. The *constant row weight* constraint makes sure that each row of Φ contains exactly L non-zero entries. The row weight L can be chosen based on signal properties such as sparsity, possible measurement noise, and details of the decoding process. Another option is to use a *constant column weight* constraint, which fixes the number of non-zero entries in each column of Φ to be a constant R.

With a constant row weight L, noisy measurement of a strictly sparse signal is approximated by two-state mixture Gaussians. To see this, consider a strictly sparse x with sparsity rate S and Gaussian variance σ_1^2. We now have $y = \Phi x + z$, where $z \sim \mathcal{N}(0, Z)$ is *additive white Gaussian noise* (AWGN) with variance Z. In our approximately sparse setting, each row of Φ picks up $\approx L(1 - S)$ small magnitude coefficients. If $L(1 - S)\sigma_0^2 \approx Z$, then the few large coefficients will be obscured by similar noise artifacts.

Our definition of Φ relies on the implicit assumption that x is sparse in the canonical sparsifying basis, i.e., $\Psi = I$. In contrast, if x is sparse in some other basis, then more complicated encoding matrices may be necessary. We defer the discussion of these issues to Section 7, but emphasize that in many practical situations our methods can be extended to support the sparsifying basis Ψ in a computationally tractable manner.

Information content of sparsely encoded measurements: The sparsity of our CS-LDPC matrix may yield measurements y that contain less information about the signal x than a fully Gaussian Φ. The following theorem, whose proof appears in Appendix A, verifies that y retains enough information to decode x well.

---

[3]CS-LDPC matrices are slightly different from LDPC parity check matrices, which only contain the binary entries 0 and 1. We have observed numerically that allowing negative entries offers improved performance. At the expense of additional computation, further minor improvement can be attained using sparse matrices with Gaussian non-zeros.

Theorem 1 *Let Φ be a CS-LDPC matrix with constant row weight $L = \eta \frac{N \ln(KN^\gamma)}{K}$, where $\eta, \gamma > 0$, and let x be a two-state mixture Gaussian signal with sparsity rate S and variances σ_0^2 and σ_1^2. If*

$$M \geq 96 \left(1 + \eta^{-1}\right) \left(1 + \gamma\right) \left(\frac{K}{\mu^2} + (N - K)\frac{\sigma_0^2}{\mu^2 \sigma_1^2}\right) \log_2(N),$$

then x can be decoded to $\widehat{x}$ such that $\|x - \widehat{x}\|_\infty \leq \mu\sigma_1$ with probability $1 - 2N^{-\gamma}$.

The proof of Theorem 1 partitions Φ into M_2 sub-matrices of M_1 rows each, and decodes each $\widehat{x_i}$ as a median of inner products with sub-matrices. The ℓ_∞ performance guarantee relies on the union bound; a less stringent guarantee yields a reduction in M_2. Moreover, L can be reduced if we increase the number of measurements accordingly. Based on numerical results, we propose the following modified values as rules of thumb,

$$L \approx S^{-1} = N/K, \quad M = O(K \log(N)), \quad \text{and} \quad R = LM/N = O(\log(N)). \tag{2}$$

Noting that each measurement requires $O(L)$ additions and subtractions, and using our rules of thumb for L and M (2), the computation required for encoding is $O(LM) = O(N \log(N))$, rather than $O(MN) = O(KN \log(N))$ for fully Gaussian Φ.

4   Decoding via Bayesian inference

Decoding approximately sparse signals is a Bayesian inference problem. We observe the measurements $y = \Phi x$, where x is a mixture Gaussian signal. Our goal is to estimate x given Φ and y. The set of equations $y = \Phi x$ is under-determined, and so there are infinitely many solutions. All solutions lie along a hyperplane of dimension $N - M$. We locate the solution within this hyperplane that best matches our prior signal model. Consider the *maximum a posteriori* (MAP) and *minimum mean square error* (MMSE) estimates,

$$\widehat{x}_{\text{MMSE}} = \arg \min_{x'} E\|X - x'\|_2^2 \quad \text{s.t.} \quad y = \Phi x',$$
$$\widehat{x}_{\text{MAP}} = \arg \max_{x'} f(X = x') \quad \text{s.t.} \quad y = \Phi x',$$

where expectation is taken over the prior distribution for X. The MMSE estimate can be expressed as the conditional mean, $\widehat{x}_{\text{MMSE}} = E[X|Y = y]$, where $Y \in \mathbb{R}^M$ is the random vector that corresponds to the measurements.

To compute the MAP and MMSE estimates of the signal x, we first consider the simple case in which we know the state configuration vector q. In this setting, x is an independently (but not identically) distributed Gaussian vector, and the MAP and MMSE estimates can be computed using the pseudo-inverse of Φ while accounting for the covariance matrix Σ_q associated with a state configuration vector q,

$$\Sigma_q = E[XX^T|Q = q],$$

where $(\cdot)^T$ denotes the transpose operator. The following theorem, which is proved in Appendix B, describes the MAP and MMSE estimators.

6

Theorem 2 *Given the measurements $y = \Phi x$ and state configuration vector q, the MAP and MMSE estimates of x coincide and are given by*

$$\widehat{x}_{\text{MAP},q} = \widehat{x}_{\text{MMSE},q} = \Sigma_q \Phi^T \left(\Phi \Sigma_q \Phi^T \right)^{-1} y. \tag{3}$$

We now present a method to compute $\widehat{x}_{\text{MMSE}}$ when the states are unknown, by relating the overall MMSE estimate to conditional MMSE estimates.

Corollary 1 *The MMSE estimate of x when the state configuration vector q is unknown is given by*

$$\widehat{x}_{\text{MMSE}} = \sum_{q \in [0,1]^N} \text{Pr}(Q = q | Y = y) \widehat{x}_{\text{MMSE},q}.$$

Proof: The expected value of the MMSE error obeys

$$
\begin{aligned}
\widehat{x}_{\text{MMSE}} &= E\left[X | Y = y\right] \\
&= \sum_{q \in [0,1]^N} \text{Pr}(Q = q | Y = y) E[X | Q = q, Y = y] \\
&= \sum_{q \in [0,1]^N} \text{Pr}(Q = q | Y = y) \widehat{x}_{\text{MMSE},q},
\end{aligned}
$$

where the last equality holds, because the MMSE estimator is the conditional mean. $\quad\square$

Computation of $\text{Pr}(Q = q | Y = y)$ relies on $f(Y = y | Q = q)$, which requires integration of $f(X = x | Q = q)$ (9) over the hyperplane of solutions to $y = \Phi x$. Taking into account the 2^N possible values for q, the practical value of Corollary 1 seems minor. Indeed, exact inference in arbitrary graphical models is NP-hard [40], because of loops in the graph induced by Φ. However, the sparse structure of Φ reduces the number of loops and enables us to use low-complexity message-passing methods to estimate x approximately.

5 CS-BP decoding of approximately sparse signals

5.1 Decoding algorithm

We now employ belief propagation (BP), an efficient method for solving inference problems by iteratively passing messages over graphical models [24–30]. BP relies on *factor graphs*, which enable fast computation of global multivariate functions by exploiting the way in which the global function factors into a product of simpler local functions, each of which depends on a subset of variables [41].

Factor graph for CS-BP: The factor graph shown in Figure 2 captures the relationship between the states q, signal coefficients x, and the observed CS measurements y. The graph is bipartite and contains two types of vertices; all edges connect *variable* nodes (black) and *constraint* nodes (white). There are three types of variable nodes corresponding to *state* variables $Q(i)$, *coefficient* variables $X(i)$, and *measurement* variables $Y(j)$. The factor graph also has three types of constraint nodes, which encapsulate the dependencies that
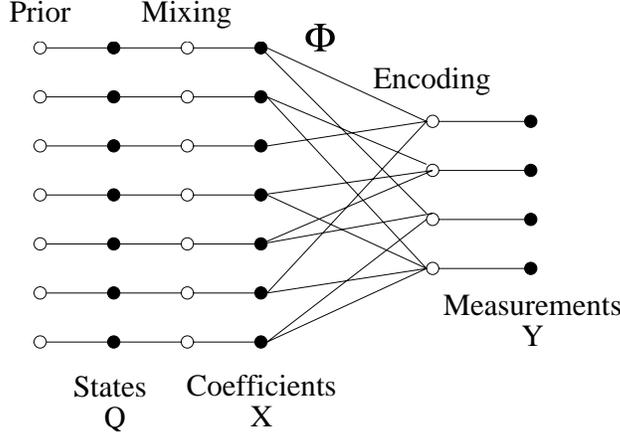
Figure 2: *Factor graph depicting the relationship between variable nodes (black) and constraint nodes (white) in CS-BP.*

their neighbors in the graph (variable nodes) are subjected to. First, *prior constraint nodes* impose the Bernoulli prior on state variables. Second, *mixing* constraint nodes impose the conditional distribution on coefficient variables given the state variables. Third, *encoding* constraint nodes impose the encoding matrix structure on measurement variables.

Message passing: CS-BP approximates the marginal distributions of all coefficient and state variables in the factor graph, conditioned on the observed measurements Y, by passing messages. Each message encodes the marginal distributions of a variable associated with one of the edges. Given the distributions $\Pr(Q(i)|Y = y)$ and $f(X(i)|Y = y)$, one can extract MAP and MMSE estimates for each coefficient. We review the basic message passing and message processing operations involved in CS-BP [26, 42].

Denote the message sent from a variable node v to one of its neighbors, a constraint node c, by $\mu_{v \longrightarrow c}(v)$; a message from c to v is denoted by $\mu_{c \longrightarrow v}(v)$. The message $\mu_{v \longrightarrow c}(v)$ is updated by taking the product of all messages received by v on all other edges. The message $\mu_{c \longrightarrow v}(v)$ is computed in a similar manner, but the constraint associated with c is applied to the product and the result is marginalized. More formally,

$$\mu_{v \longrightarrow c}(v) = \prod_{u \in n(v) \backslash \{c\}} \mu_{u \longrightarrow v}(v), \tag{4}$$

$$\mu_{c \longrightarrow v}(v) = \sum_{\sim \{v\}} \left(\text{con}(n(c)) \prod_{w \in n(c) \backslash \{v\}} \mu_{w \longrightarrow c}(w) \right), \tag{5}$$

where $n(v)$ and $n(c)$ are sets of neighbors of v and c, respectively, $\text{con}(n(c))$ is the constraint on the set of variable nodes $n(c)$, and $\sim \{v\}$ is the set of neighbors of c excluding v. We interpret these 2 types of message processing as *multiplication* of beliefs at variable nodes (4) and *convolution* at constraint nodes (5). Finally, the marginal distribution $f(v)$ for a given variable node is obtained from the product of all the most recent incoming messages along the edges connecting to that node,

$$f(v) = \prod_{u \in n(v)} \mu_{u \longrightarrow v}(v). \tag{6}$$

Based on the marginal distribution, various statistical characterizations can be computed, including MMSE, MAP, error bars, and so on.

We also need a method to encode beliefs. One method is to sample pdf's uniformly and use the samples as messages. Another encoding method is to approximate the pdf by a mixture Gaussian with a given number of components, where mixture parameters are used as messages. These two methods offer different tradeoffs between modeling flexibility and computational requirements; details appear in Sections 5.2 and 5.3. We leave alternative methods such as particle filters and importance sampling for future research.

Protecting against loopy graphs and message quantization errors: BP converges to the exact conditional distribution in the ideal situation where the following conditions are met: (i) the factor graph is cycle-free; and (ii) messages are processed and propagated without errors. In CS-BP decoding, both conditions are violated. First, the factor graph is loopy – it contains cycles. Second, message encoding methods introduce errors. These non-idealities may lead CS-BP to converge to imprecise conditional distributions, or more critically, lead CS-BP to diverge [43–45]. To some extent these problems can be reduced by (i) using CS-LDPC matrices, which have a relatively modest number of loops; and (ii) carefully designing our message encoding methods (Sections 5.2 and 5.3). We use message damped belief propagation (MDBP) [46] to stabilize CS-BP against these non-idealities. We conclude with a prototype algorithm; Matlab code is available at http://dsp.rice.edu/∼drorb/CSBP.

<div align="center">

CS-BP Decoding Algorithm

</div>

1. **Initialization**: Initialize the iteration counter $i = 1$. Set up data structures for factor graph messages $\mu_{v \longrightarrow c}(v)$ and $\mu_{c \longrightarrow v}(v)$. Initialize messages $\mu_{v \longrightarrow c}(v)$ from variable to constraint nodes with the signal prior.

2. **Convolution**: For each measurement $c = 1, \ldots, M$, which corresponds to constraint node c, compute $\mu_{c \longrightarrow v}(v)$ via convolution (5) for all neighboring variable nodes $n(c)$. If measurement noise is present, then convolve further with a noise prior. Apply damping methods such as MDBP [46] to stabilize CS-BP against non-idealities.

3. **Multiplication**: For each coefficient $v = 1, \ldots, N$, which corresponds to a variable node v, compute $\mu_{v \longrightarrow c}(v)$ via multiplication (4) for all neighboring constraint nodes $n(v)$. Apply damping methods as needed. If the iteration counter has yet to reach its maximal value, then go to Step 2.

4. **Output**: For each coefficient $v = 1, \ldots, N$, compute MMSE or MAP estimates (or alternative statistical characterizations) based on the marginal distribution $f(v)$ (6). Output the requisite statistics.

5.2   Samples of the pdf as messages

Having described main aspects of the CS-BP decoding algorithm, we now focus on the two message encoding methods, starting with samples. In this method, we sample the pdf and send the samples as messages. Multiplication of pdf's (4) corresponds to point-wise multiplication of messages; convolution (5) is computed efficiently in the frequency domain.[4]

---

[4]Fast convolution via FFT has been used in LDPC decoding over $GF(2^q)$ using BP [29].

The main advantage of using samples is flexibility to different prior distributions for the coefficients; for example, mixture Gaussian priors are easily supported. Additionally, both multiplication and convolution are computed efficiently. However, sampling has large memory requirements and introduces quantization errors that reduce precision and hamper the convergence of CS-BP [43]. Sampling also requires finer sampling for precise decoding; we propose to sample the pdf's with a spacing less than σ_0.

We analyze the computational requirements of this method. Let each message be a vector of p samples. Each iteration performs multiplication at coefficient nodes (4) and convolution at constraint nodes (5). Outgoing message are modified,

$$\mu_{v\longrightarrow c}(v) = \frac{\prod_{u\in n(v)} \mu_{u\longrightarrow v}(v)}{\mu_{c\longrightarrow v}(v)} \quad \text{and} \quad \mu_{c\longrightarrow v}(v) = \sum_{\sim\{v\}}\left(\text{con}(n(c))\frac{\prod_{w\in n(c)} \mu_{w\longrightarrow c}(w)}{\mu_{v\longrightarrow c}(v)}\right), \quad (7)$$

where the denominators are non-zero, because mixture Gaussian pdf's are strictly positive. The modifications (7) reduce computation, because the numerators are computed once and then reused for all messages leaving the node being processed.

Assuming that the column weight R is fixed (Section 3), the computation required for message processing at a variable node is $O(Rp)$ per iteration, because we multiply $R+1$ vectors of length p. With $O(N)$ variable nodes, each iteration is $O(NRp)$. For constraint nodes, we perform convolution in the frequency domain, and so the computational cost per node is $O(Lp\log(p))$. With $O(M)$ constraint nodes, each iteration is $O(LMp\log(p))$. Accounting for both variable and constraint nodes, each iteration is $O(NRp+LMp\log(p)) = O(p\log(p)N\log(N))$, where we employ our rules of thumb for L, M, and R (2). To complete the computational analysis, we note first that we use $O(\log(N))$ CS-BP iterations, which is proportional to the diameter of the graph [42]. Second, sampling the pdf's with a spacing less than σ_0, we choose $p = O(\sigma_1/\sigma_0)$ to support a dispersion of σ_1. Therefore, our overall computation is $O(\frac{\sigma_1}{\sigma_0}\log\left(\frac{\sigma_1}{\sigma_0}\right)N\log^2(N))$, which scales as $O(N\log^2(N))$.

5.3 Mixture Gaussian parameters as messages

In this method, we approximate the pdf by a mixture Gaussian with a maximum number of components, and then send the mixture parameters as messages. For both multiplication (4) and convolution (5), the resulting number of components in the mixture is multiplicative in the number of constituent components. To keep the message representation tractable, we use *Iterative Pairwise Replacement Algorithm* (IPRA) for model order reduction.

The advantage of using mixture Gaussians to encode pdf's is that the messages are short and hence consume little memory. This method works well for mixture Gaussian priors, but could be difficult to adapt to other priors. Model order reduction algorithms such as IPRA can be computationally expensive [47], and introduce errors in the messages, which impair the quality of the solution as well as the convergence of CS-BP [43].

Again, we analyze the computational requirements. Because it is impossible to undo the multiplication in (4) and (5), we cannot use the modified form (7). Let m be the maximum model order. Model order reduction using IPRA [47] requires $O(m^2R^2)$ computation per coefficient node per iteration. With $O(N)$ coefficient nodes, each iteration is $O(m^2R^2N)$. Sim-

Table 1: *Computational and storage requirements of CS-BP decoding*

Messages	Parameter	Computation	Storage
Samples of pdf	p samples	$O(\frac{\sigma_1}{\sigma_0}\log\left(\frac{\sigma_1}{\sigma_0}\right)N\log^2(N))$	$O(pN\log(N))$
Mixture Gaussians	m components	$O(m^2\frac{N}{S}\log^2(N))$	$O(mN\log(N))$

ilarly, with $O(M)$ constraint nodes, each iteration is $O(m^2L^2M)$. Accounting for $O(\log(N))$ CS-BP iterations, overall computation is $O(m^2[L^2M + R^2N]\log(N)) = O(m^2\frac{N}{S}\log^2(N))$.

5.4   Properties of CS-BP decoding

We briefly describe several properties of CS-BP decoding. The two methods for encoding beliefs about conditional distributions have different *computational characteristics*, and were evaluated in Sections 5.2 and 5.3. The *storage requirements* are mainly for message representation of the $LM = O(N\log(N))$ edges. For encoding with pdf samples, the message length is p, and so the storage requirement is $O(pN\log(N))$. For encoding with mixture Gaussian parameters, the message length is m, and so the storage requirement is $O(mN\log(N))$. Computational and storage requirements are summarized in Table 1.

   Several additional properties are now featured. First, we have *progressive decoding*; more measurements will improve the precision of the estimated posterior probabilities. Second, if we are only interested in the state configuration vector q but not in the coefficient values, then we need fewer measurements, because we need to extract less information from the measurements. We call this *information scalability*, and in some applications it can result in drastic resource conservation. Third, we have *robustness to noise*, because noisy measurements can be incorporated into our model by convolving the noiseless version of the estimated pdf (5) at each encoding node with the pdf of the noise.

6   Numerical results

To demonstrate the efficacy of CS-BP, we simulated decoding problems where $N = 1000$, $S = 0.1$, $\sigma_1 = 10$, $\sigma_0 = 1$, and the measurements are noiseless. We used samples of the pdf as messages, where each message consisted of $p = 243 = 3^5$ samples; this choice of p provided fast FFT computation. Figure 3 plots the MMSE decoding error as a function of M for a variety of row weights L. The figure emphasizes with dashed lines the average ℓ_2 norm of x (top) and of the small coefficients (bottom); increasing M reduces the decoding error, until it reaches the energy level of the small coefficients. A small row weight, e.g., $L = 5$, may miss some of the large coefficients and is thus bad for decoding; as we increase L, fewer measurements are needed to obtain the same precision. However, there is an optimal $L_{\text{opt}} \approx 2/S = 20$ beyond which any performance gains are marginal. Furthermore, values of $L > L_{\text{opt}}$ give rise to divergence in CS-BP, even with damping.

   To compare the performance of CS-BP to ℓ_1 decoding based on linear programming (LP) and CoSaMP [14], a fast greedy solver, we simulated all three methods where $N = 500$, $S = 0.1$, $L = 20$, $\sigma_1 = 10$, $\sigma_0 = 1$, $p = 243$, and the measurements are noiseless. In order for
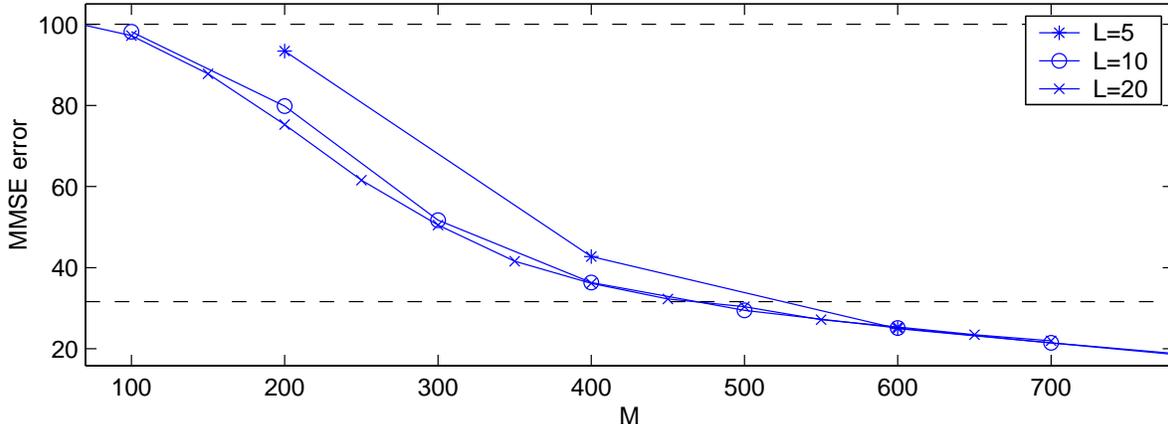
Figure 3: *MMSE error as a function of the number of measurements M using different matrix row weights L. The dashed lines show the ℓ_2 norms of x (top) and the small coefficients (bottom). ($N = 1000$, $S = 0.1$, $\sigma_1 = 10$, $\sigma_0 = 1$, and noiseless measurements.)*

the comparison to be as fair as possible, throughout the experiment we used the same CS-LDPC encoding matrix Φ, the same signal x, and same measurements y. Figure 4 plots the MMSE decoding error as a function of M for CS-BP, ℓ_1, and CoSaMP decoding. For small M, CS-BP exploits its knowledge about the approximately sparse structure of x, and has a smaller decoding error. CS-BP requires $20 - 30\%$ fewer measurements than LP to obtain the same MMSE decoding error; the advantage over CoSaMP is even greater. However, as M increases, the advantage of CS-BP over ℓ_1 decoding becomes less pronounced.

To demonstrate that CS-BP deals well with measurement noise, recall the noisy measurement setting $y = \Phi x + z$ of Section 3, where $z \sim \mathcal{N}(0, Z)$ is AWGN with variance Z. Our algorithm deals with noise by convolving the noiseless version of the estimated pdf (5) with the noise pdf. We simulated decoding problems where $N = 1000$, $S = 0.1$, $L = 20$, $\sigma_1 = 10$, $\sigma_0 = 1$, $p = 243$, and $Z \in \{0, 2, 5, 10\}$. Figure 5 plots the MMSE decoding error as a function of M and Z. To put things in perspective, the average measurement picks up a Gaussian term of variance $L(1 - S)\sigma_0^2 = 18$ from the signal. Although the decoding error increases with Z, as long as $Z \ll 18$ the noise has little impact on the decoding error; CS-BP offers a graceful degradation to measurement noise.

Our final experiment considers the situation where CS-BP has an imprecise statistical characterization of the signal. Instead of having a two-state mixture Gaussian signal model as before, where large coefficients have variance σ_1^2 and occur with probability S, we defined a C-state mixture model. In our definition, σ_0^2 is interpreted as a background signal level, which appears in all coefficients. Whereas the two-state model adds a "true signal" component of variance $\sigma_1^2 - \sigma_0^2$ to the background signal, the $C - 1$ large components each occur with probability S and the amplitudes of the true signals are $\sigma_2, 2\sigma_2, \ldots, (C - 1)\sigma_2$, where σ_2 is chosen to preserve the total signal energy. At the same time, we did not change the signal priors in CS-BP, and used the same two-state mixture model as before. We simulated decoding problems where $N = 1000$, $S = 0.1$, $L = 20$, $\sigma_1 = 10$, $\sigma_0 = 1$, $p = 243$, the measurements are noiseless, and $C \in \{2, 3, 4, 5\}$. Figure 6 plots the MMSE decoding error as a function of M and C. As the number of mixture components C increases, the MMSE
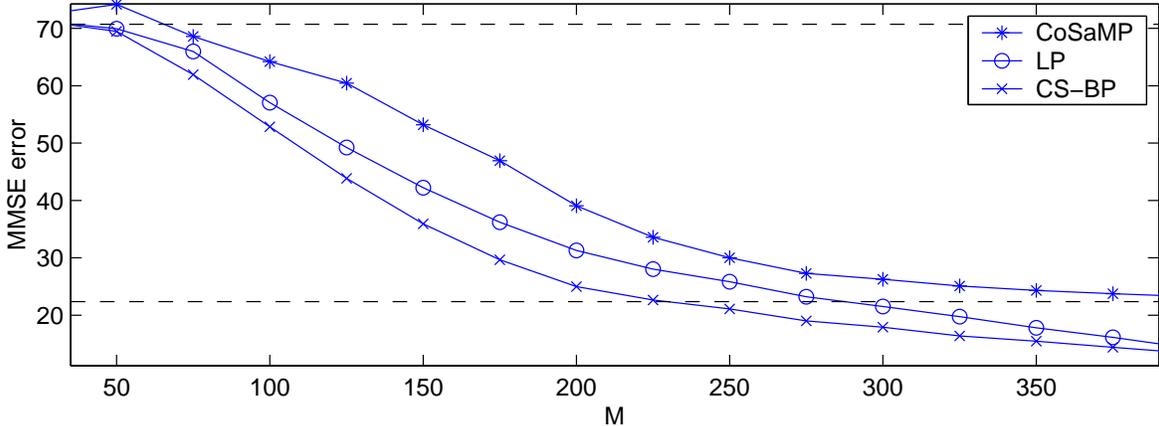
12

Figure 4: *MMSE error as a function of the number of measurements M using CS-BP, linear programming (LP) ℓ_1 decoding, and CoSaMP. The dashed lines show the ℓ_2 norms of x (top) and the small coefficients (bottom). ($N = 500$, $S = 0.1$, $L = 20$, $\sigma_1 = 10$, $\sigma_0 = 1$, and noiseless measurements.)*

error increases. However, even for $C = 3$ the sparsity rate effectively doubles from S to $2S$, and an increase in the required number of measurements M is expected.

7   Variations and enhancements

Supporting arbitrary sparsifying basis Ψ: Until now, we assumed that the canonical sparsifying basis, i.e., $\Psi = I$, is used. In this case, x itself is sparse. We now explain how CS-BP can be modified to support the case where x is sparse in an arbitrary basis Ψ. In the encoder, we multiply the CS-LDPC matrix Φ by Ψ^T and encode x as $y = (\Phi\Psi^T)x = (\Phi\Psi^T)(\Psi\theta) = \Phi\theta$. In the decoder, we use BP to form the approximation $\widehat{\theta}$, and then transform via Ψ to $\widehat{x} = \Psi\widehat{\theta}$. In order to construct the modified encoding matrix $\Phi\Psi^T$ and later transform $\widehat{\theta}$ to $\widehat{x}$, extra computation is needed; this extra cost is $O(N^2)$ in general. Fortunately, in many practical situations Ψ is structured (e.g., Fourier or wavelet bases) and amenable to fast computation. Therefore, extending our methods to such bases is feasible.

Exploiting statistical dependencies: In many signal representations, the coefficients are not iid. For example, wavelet representations of natural images often contain correlations between magnitudes of parent and child coefficients [2]. Consequently, it is possible to decode signals from fewer measurements using an algorithm that allocates different distributions to different coefficients [38, 48]. By modifying the dependencies imposed by the prior constraint nodes (Section 5.1), CS-BP decoding supports different signal models.

Feedback: Feedback from the decoder to the encoder can be used in applications where measurements may be lost because of transmissions over faulty channels. In an analogous manner to a digital fountain [49], the marginal distributions (6) enable to identify when sufficient information for signal decoding has been received. At that stage, the decoder notifies the encoder that decoding is complete, and measurements are no longer streamed.

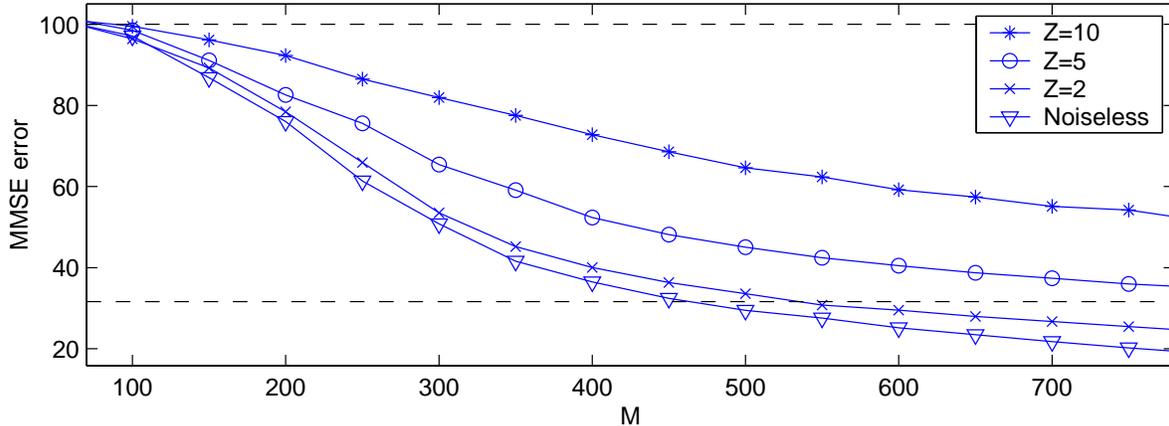**Irregular CS-LDPC matrices**: In channel coding, LDPC matrices that have irregular

Figure 5: *MMSE error as a function of M using different noise levels Z. The dashed lines show the ℓ_2 norms of x (top) and the small coefficients (bottom). ($N = 1000$, $S = 0.1$, $L = 20$, $\sigma_1 = 10$, and $\sigma_0 = 1$.)*

row and column weights come closer to the Shannon limit, because a small number of rows or columns with large weights require only modest additional computation yet greatly reduce the block error rate [32]. In an analogous manner, we expect irregular CS-LDPC matrices to enable a further reduction in the number of measurements required.

8 Discussion

This paper uses a sparse encoding matrix and belief propagation to accelerate CS encoding and decoding under the Bayesian framework. Although we focus on decoding approximately sparse signals, CS-BP can be extended to signals that are sparse in other bases, is flexible to modifications in the signal model, and can address measurement noise.

Despite the significant benefits, CS-BP is not universal in the sense that it does not work for signals that are sparse in any basis. Nonetheless, our framework can be applied to arbitrary bases by modifying the encoding matrix and decoding methods.

Our method resembles low density parity check (LDPC) codes [31, 32], which use a sparse Bernoulli parity check matrix. Although any linear code can be represented as a bipartite graph, for LDPC codes the sparsity of the graph accelerates the encoding and decoding processes. LDPC codes are celebrated for achieving rates close to the Shannon limit. A similar comparison of the MMSE performance of CS-BP with information theoretic bounds on CS performance is left for future research.

In comparison to previous work on Bayesian aspects of CS [22, 23], our method is much faster, requiring only $O(N \log^2(N))$ computation. At the same time, CS-BP offers significant flexibility, and should not be viewed as merely another fast CS decoding algorithm [10–17, 19–21, 33, 34]. However, CS-BP relies on the sparsity of CS-LDPC matrices, and future research can consider the applicability of such matrices in different applications.
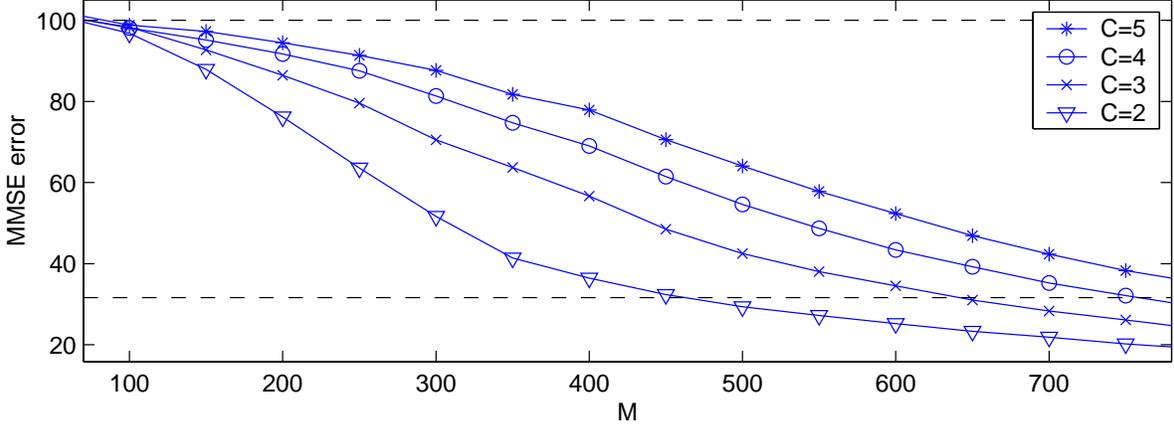
Figure 6: *MMSE error as a function of the number of measurements M and the number of states C in the mixture Gaussian signal model. The dashed lines show the ℓ_2 norms of x (top) and the small coefficients (bottom). ($N = 1000$, $S = 0.1$, $L = 20$, $\sigma_1 = 10$, $\sigma_0 = 1$, and noiseless measurements.)*

Appendix A   Proof of Theorem 1

We first review a result by Wang et al. [9, Theorem 1] and then describe its application to Theorem 1. Wang et al. prove for an input $x \in \mathbb{R}^N$ and collection of N test vectors $v_i \in \mathbb{R}^N$ that $M = O(\frac{1}{\epsilon^2} \max\{1, sQ^2\} \log(N))$ measurements suffice to estimate $x^T v_i$ up to resolution $\epsilon \|x\|_2 \|v\|_2$, where $1/s$ is the fraction of non-zero entries in Φ, and $Q \geq \frac{\|x\|_\infty}{\|x\|_2}$. The proof partitions Φ into M_2 sub-matrices of M_1 rows each. The authors define $w_i = \frac{1}{\sqrt{M_1}}\Phi_i x$, $y_i = \frac{1}{\sqrt{M_1}}\Phi_i v$, and $z_i = (w_i)^T y_i$, where $i \in \{1, ..., M_2\}$, and v can be any of the N test vectors. The event $\xi_i \triangleq 1_{\{|z_i - x^T v| > \epsilon \|x\|_2 \|v\|_2\}}$, where $1_{\{\cdot\}}$ is an indicator function, identifies that z_i lies outside a *tolerable interval* of $x^T v$. Wang et al. prove that $\Pr(\xi_i = 1) \leq (2+sQ^2)/(\epsilon^2 M_1) \triangleq p$. Assigning $M_1 = 4(2 + sQ^2)/\epsilon^2$ yields $p = \frac{1}{4}$. Next, the approximation $\widehat{v}$ is defined as the median of $\{z_1, \ldots, z_{M_2}\}$. As long as M_2 is sufficiently large, $\widehat{v}$ is within the tolerable interval with arbitrarily high probability. Indeed, Wang et al. show [9] that

$$\Pr\left(\sum_{i=1}^{M_2} \xi_i > (1 + c)\frac{M_2}{4}\right) < e^{-\frac{c^2 M_2}{12}},$$

where $0 < c < 1$ is some constant. Choosing $c = 1$ ensures that the majority of ξ_i are zero, and so $\widehat{v}$ must lie within the tolerable interval. Assigning $M_2 = 12(1 + \gamma) \log_2(N)$ for some $\gamma > 0$, the median algorithm approximates v by $\widehat{v}$ within the tolerable interval with probability $1 - N^{-(1+\gamma)}$. Considering a union bound over all N test vectors v_i, all N medians fall within the tolerable interval with probability lower bounded by $1 - N^{-\gamma}$.

Application to Theorem 1: In our problem, $\|x\|_2 \approx \sqrt{K\sigma_1^2 + (N - K)\sigma_0^2}$. The maximum among J unit norm Gaussian random variables has amplitude close to $\sqrt{2 \ln(J)}$, and so $\|x\|_\infty < \sqrt{2 \ln(KN^\gamma)}\sigma_1$ with probability $1 - N^{-\gamma}$. When this inequality holds, we have

$$\frac{\|x\|_\infty}{\|x\|_2} < \frac{\sqrt{2 \ln(KN^\gamma)}\sigma_1}{\sqrt{K\sigma_1^2 + (N - K)\sigma_0^2}} < \frac{\sqrt{2 \ln(KN^\gamma)}\sigma_1}{\sqrt{K\sigma_1^2}} = \sqrt{\frac{2 \ln(KN^\gamma)}{K}}, \tag{8}$$

which suggests $Q = \sqrt{2\ln(KN^\gamma)/K}$. Next, we assign $s = \frac{K}{\eta \ln(KN^\gamma)}$ to ensure $N/s = \eta N \ln(KN^\gamma)/K$ non-zero Φ entries per row. To decode x, choose $(v_i)_{i=1}^N$ to be the N canonical vectors of the identity matrix I_N. We have $x^T v_i = x_i$, and approximate $\widehat{x}_i$ by the median algorithm. With probability lower bounded by $1 - N^{-\gamma}$, all N approximations satisfy $|x_i - \widehat{x}_i| \leq \epsilon \|x\|_2 \|v\|_2 \approx \epsilon \sqrt{K\sigma_1^2 + (N-K)\sigma_0^2}$. In order for $\|x - \widehat{x}\|_\infty \leq \mu\sigma_1$, we require $\epsilon = \mu\sigma_1/\sqrt{K\sigma_1^2 + (N-K)\sigma_0^2}$. Finally,

$$
\begin{aligned}
M &= M_1 M_2 = \left(\frac{4(2 + sQ^2)}{\epsilon^2} \right) (12(1+\gamma)\log_2(N)) \\
&= 96\left(1 + \eta^{-1}\right)(1+\gamma)\left(\frac{K}{\mu^2} + (N-K)\frac{\sigma_0^2}{\mu^2\sigma_1^2} \right)\log_2(N).
\end{aligned}
$$

We complete the proof by noting that large $\|x\|_\infty$ (8) or large $\|x - \widehat{x}\|_\infty$ each occur with probability $N^{-\gamma}$. $\qquad\square$

Appendix B Proof of Theorem 2

Conditioned on knowing the state configuration vector q, the distribution of X is a multivariate Gaussian with covariance Σ_q. Therefore,

$$
f\left(X = x | Q = q\right) = \frac{1}{(2\pi)^{N/2}\det(\Sigma_q)^{1/2}} e^{-\frac{1}{2}x^T \Sigma_q^{-1} x}. \tag{9}
$$

The probability $f(x|q)$ is maximized when $x^T \Sigma_q^{-1} x$ is minimum. Consider the MAP estimate, $\widehat{x}_{\mathrm{MAP},q} = \arg\min_{x'} (x')^T \Sigma_q^{-1} x'$ s.t. $y = \Phi x'$. The covariance matrix Σ_q is diagonal, because x is zero mean iid. If $q(j) = 1$, then the j'th diagonal entry is equal to σ_1^2; else the diagonal entry is σ_0^2. Because all diagonal entries are positive, Σ_q is invertible. Consequently, we can consider the change of variables, $w = \Sigma_q^{-1/2} x'$. We now have $w_{\mathrm{MAP}} = \arg\min_w w^T w$ s.t. $y = \Phi\Sigma_q^{1/2}w$. This least squares problem can be solved using the pseudo-inverse,

$$
w_{\mathrm{MAP}} = (\Phi\Sigma_q^{1/2})^+ y = \Sigma_q^{1/2}\Phi^T \left(\Phi\Sigma_q\Phi^T\right)^{-1} y.
$$

Because $\widehat{x}_{\mathrm{MAP},q} = \Sigma_q^{1/2} w_{\mathrm{MAP}}$, we have proved the result (3) for $\widehat{x}_{\mathrm{MAP},q}$. We complete the proof by noting that $\widehat{x}_{\mathrm{MMSE},q} = \widehat{x}_{\mathrm{MAP},q}$ for multivariate Gaussian random variables. $\qquad\square$

Acknowledgments

References

[1] S. Sarvotham, D. Baron, and R. G. Baraniuk, "Compressed sensing reconstruction via belief propagation," Tech. Rep. TREE0601, Rice University, Houston, TX, July 2006.

[2] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 719–746, Mar. 1992.

[3] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[4] D. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] E. Candès, M. Rudelson, T. Tao, and R. Vershynin, "Error correction via linear programming," *Found. of Comp. Math.*, pp. 295–308, 2005.

[6] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Jour. Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[7] D. Donoho and J. Tanner, "Neighborliness of randomly projected simplices in high dimensions," *Proc. National Academy of Sciences*, vol. 102, no. 27, pp. 9452–457, 2005.

[8] D. Donoho, "High-dimensional centrally symmetric polytopes with neighborliness proportional to dimension," *Discrete and Computational Geometry*, vol. 35, no. 4, pp. 617–652, Mar. 2006.

[9] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *Proc. Information Processing Sensor Networks (IPSN2007)*, 2007, pp. 331–339.

[10] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[11] A. Cohen, W. Dahmen, and R. A. DeVore, "Near optimal approximation of arbitrary vectors from highly incomplete measurements," 2007, Preprint.

[12] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk, "Fast reconstruction of piecewise smooth signals from random projections," in *Proc. SPARS05*, Rennes, France, Nov. 2005.

[13] D. L. Donoho, Y. Tsaig, I. Drori, and J-C Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," Mar. 2006, Preprint.

[14] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, 2008, To appear.

[15] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing: Closing the gap between performance and complexity," 2008, Submitted.

[16] E. Hale, W. Yin, and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," 2007, Submitted.

[17] M. Figueiredo, R. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," Feb., IEEE Jour. Selected Topics Signal Processing.

[18] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," Tech. Rep., Department of Computer Science, University of British Columbia.

[19] G. Cormode and S. Muthukrishnan, "Towards an algorithmic theory of compressed sensing," *DIMACS Technical Report TR 2005-25*, 2005.

[20] G. Cormode and S. Muthukrishnan, "Combinatorial algorithms for compressed sensing," *DIMACS Technical Report TR 2005-40*, 2005.

[21] A. C. Gilbert, M. J. Strauss, J. Tropp, and R. Vershynin, "Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors," Apr. 2006, Submitted.

[22] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Trans. Signal Processing*, vol. 56, no. 6, pp. 2346–2356, June 2008.

[23] M. W. Seeger and H. Nickisch, "Compressed sensing and Bayesian experimental design," in *ICML '08: Proc. 25th International Conference on Machine learning*, 2008, pp. 912–919.

[24] J. Pearl, "Probablistic reasoning in intelligent systems: Networks of plausible inference," *Morgan-Kaufmann*, 1988.

[25] F. V. Jensen, "An introduction to Bayesian networks," *Springer-Verlag*, 1996.

[26] B. J. Frey, "Graphical models for machine learning and digital communication," *MIT press*, 1998.

[27] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Mitsubishi Tech. Rep. TR2001-022*, Jan. 2002.

[28] F. V. Jensen, "Bayesian networks and decision graphs," *Springer-Verlag*, May 2002.

[29] D. J. C. MacKay, "Information theory, inference and learning algorithms," *Cambridge University Press*, 2002, ISBN 0521642981.

[30] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, "Probabilistic networks and expert systems," *Springer-Verlag*, 2003.

[31] R. G. Gallager, "Low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 8, pp. 21–28, Jan. 1962.

[32] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[33] S. Sarvotham, D. Baron, and R. G. Baraniuk, "Sudocodes – Fast measurement and reconstruction of sparse signals," in *Proc. International Symposium Information Theory (ISIT2006)*, Seattle, WA, July 2006.

[34] R. Berinde and P. Indyk, "Sparse recovery using sparse random matrices," *MIT-CSAIL-TR-2008-001*, 2008, Technical Report.

[35] J.-C Pesquet, H. Krim, and E. Hamman, "Bayesian approach to best basis selection," *IEEE 1996 Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, pp. 2634–2637, 1996.

[36] H. Chipman, E. Kolaczyk, and R. McCulloch, "Adaptive Bayesian wavelet shrinkage," *Jour. Amer. Stat. Assoc.*, vol. 92, 1997.

[37] M. Crouse, R. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 46, no. 4, pp. 886–902, Apr. 1998.

[38] L. He and L. Carin, "Exploiting structure in wavelet-based Bayesian compressed sensing," 2008, Submitted.

[39] H. W. Sorenson and D. L. Alspach, "Recursive Bayesian estimation using Gaussian sums," *Automatica*, vol. 7, pp. 465–479, 1971.

[40] G. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intelligence*, vol. 42, pp. 393–405, 1990.

[41] F. R. Kschischang, B. J. Frey, and H-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[42] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, pp. 399–431, Mar. 1999.

[43] E. Sudderth, A. Ihler, W. Freeman, and A. S. Willsky, "Nonparametric belief propagation," *MIT LIDS Tech. Rep. 2551*, Oct. 2002.

[44] B. J. Frey and D. J. C. MacKay, "A revolution: Belief propagation in graph with cycles," *Adv. in Neural Information Processing Systems, M. Jordan, M. S. Kearns and S. A. Solla (Eds.)*, vol. 10, 1998.

[45] A. Ihler, J. Fisher, and A. S. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *Jour. of Machine Learning Research*, vol. 6, pp. 905–936, May 2005.

[46] M. Pretti, "A Message-Passing Algorithm with Damping," *Jour. Stat. Mech.*, Nov. 2005.

[47] D. W. Scott and W. F. Szewczyk, "From kernels to mixtures," *Technometrics*, vol. 43, pp. 323–335, Aug. 2001.

[48] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," 2008, Preprint.

[49] J. W. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to asynchronous reliable multicast," *IEEE Jour. Select. Areas Commun.*, vol. 20, no. 8, pp. 1528–1540, Oct. 2002.