SPARSE SIGNAL RECONSTRUCTION FROM LIMITED DATA USING BAYESIAN COMPRESSIVE SENSING VIA BELIEF PROPAGATION

Prateek Paliwal and Manish Sharma

ABSTRACT

Compressive sensing (CS) is a novel paradigm for acquiring signals, based on the idea that one can efficiently capture all the information of sparse domain by sampling only a part of signal through sub-Nyquist signal acquisition. As Bayesian inference is substituting conventional CS methods, the work employ Bayesian inference which depict CS matrix by a factor graph to accelerate both encoding and belief propagation (BP) decoding. Two state mixture Gaussian model is used to model prior for sparse signal. To decode a length- N signal having K large coefficients, our BCS-BP decoding algorithm uses $O(K \log(N))$ and $O(N \log^2(N))$ computation. BCS-BP algorithm is easily versatile to various signal models.

Keywords: *Compressive sensing, bayesian inference, belief propagation, sparse matrices, decoding algorithm.*

Reference *to this paper should be made as follows: Prateek Palliwal and Manish Sharma, (2019), "Sparse Signal Reconstruction From Limited Data Using Bayesian Compressive Sensing Via Belief Propagation" Int. J. Electronics Engineering and Applications, Vol. 7, No. 2, pp. 24-36*

Biographical notes:

Prateek Paliwal: *2SOE, Sanghvi Institute of Management & Science, Indore (M.P), India.*

I. INTRODUCTION

Traditional approaches of sampling signals follow Shannon's theorem to construct a signal without error, the sampling rate must be at least the Nyquist rate. Traditional sensing schemes of acquiring data require larger statistical characterization of the signal for estimation and recovery. Interestingly, the information possessed by the signal in various signal processing applications requires large coefficients for computations which can be easily estimated by few small coefficients. Clearly, it is wasteful to compute all of the coefficients as most of them will be discarded later on. This led to the evolution of a Compressive sensing (CS) a technique based on the idea that one can sufficiently capture all the information in a sparse signal by sampling only part of the signal using a sampling domain that is incoherent to the signal representation domain. Compressive sensing represents a paradigm in which number of measurements is reduced during acquisition. CS relies on two principles: sparsity, which reveals to the signals of interest, and incoherence, which reveals to the sensing modality [1]. CS enables a significant reduction in the sampling and computation costs for signals having sparse representation in some basis.

1.1 Compressive Sensing

Compressed sensing is a technique to reconstruct a signal accurately and efficiently from a set of few linear measurements. Compressive sensing address inefficiencies by directly acquiring a compressed signal representation without going through the intermediate stage of acquiring N samples [2].

Sparsity and Random Encoding: Consider a signal vector $x \in \mathbb{R}^N$ that has only $K \ll N$ non-zero coefficients. The core tenet of CS is that it is unnecessary to measure all the N values of x ; rather, we can recover x from a small number of projections onto an incoherent basis [3], [4]. To measure (encode) x , we compute the measurement vector $y \in \mathbb{R}^M$ as M linear Projections of x via the matrix-vector multiplication $y = \Phi x$ where $\Phi \in \mathbb{R}^{M \times N}$ is the encoding matrix. Approximately sparse signals have large coefficients, while the remaining coefficients are small but not necessarily zero. Compressible signals have coefficients that, when sorted, decay quickly according to a power law. Similarly, both noiseless and noisy signals and measurements may be considered. We emphasize noiseless measurement of approximately sparse signals in the paper.

Decoding via Sparsity: Our goal is to reconstruct (decode) x either accurately or approximately given y and Φ using $M \ll N$ measurements. Although the recovery of the signal x from the measurements $y = \Phi x$ appears to be a severely ill-posed inverse problem, the strong prior knowledge of sparsity in x gives us hope to reconstruct x using $M \ll N$ measurements. In fact the signal recovery can be achieved using optimization by searching for the sparsest signal that agrees with the M observed measurements in y . The key observation is that the signal x is the solution to the ℓ_0 minimization

$$\hat{x} = \arg \min ||x||_0 \text{ s.t. } y = \Phi x \quad (1)$$

With overwhelming probability as long as we have sufficiently many measurements, where $||\cdot||_0$ denotes the ℓ_0 "norm" that counts the number of non-zero elements. Unfortunately, solving the ℓ_0 optimization is known to be an NP-complete problem [5]. In order to recover the signal, the decoder needs to perform combinatorial enumeration of all the ( ) possible sparse subspaces.

The practical revelation that support the CS theory is that it is not necessary to resort to combinatorial search to recover the set of non-zero coefficients of x from the ill-posed inverse

Problem $y = \Phi x$. A much easier problem yields an equivalent solution. We need only solve for the ℓ_1 -sparsest coefficients that agree with the measurements y .

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|x\|_1 \text{ s.t. } y = \Phi x \quad (2)$$

as long as Φ satisfies the restricted isometry (RIP) condition. Furthermore, the ℓ_1 optimization problem also known as Basis Pursuit [6], is significantly more approachable and can be solved with linear programming techniques.

1.2 Bayesian CS

CS decoding algorithms rely on the sparsity of the signal. 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. [7] 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 [8] extend these ideas to experimental design, where the encoding matrix is designed sequentially based on previous measurements. Another Bayesian approach by Schniter et al. [9] approximates conditional expectation by extending the maximal likelihood approach to a weighted mixture of the most likely models. There are also many related results on application of Bayesian methods to sparse inverse problems [10]. Bayesian approaches have also been used for multiuser decoding (MUD) in communications. In MUD, users modulate their symbols with different spreading sequences, and the received signals are superposition of sequences. Because most users are inactive, MUD algorithms extract information from a sparse superposition in a manner analogous to CS decoding. Guo and Wang [11] perform MUD using sparse spreading sequences and decode via belief propagation (BP) [12]; our paper also uses sparse encoding matrices and BP decoding. A related algorithm for decoding low-density lattice codes (LDLC) by Sommer et al. [13] uses BP on a factor graph whose self and edge potentials are Gaussian mixtures. Convergence results for the LDLC decoding algorithm have been derived for Gaussian noise.

1.3 Contributions

In this paper, the concentrate is on developing a sparse encoder matrix Φ and a belief propagation (BP) decoder to accelerate CS encoding and decoding under the Bayesian techniques. We call our algorithm BCS-BP. Two state mixture Gaussian model is used to model prior for sparse signal, BCS-BP is versatile in various signal and measurement models.

Encoding by Sparse CS Matrix: The dense sub-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 [14], [15] 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. 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 [16]. Although LDPC decoding algorithms may not have provable convergence, the recent extension of LDPC to LDLC codes [13] offers provable convergence. Additionally, CS-BP has recently been proved to be asymptotically optimal in the large system limit [11].

We encode (measure) the signal using sparse 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 . Based on the pertinent signal and measurement models, the layout of Φ , including characteristics such as column and row weights are estimated, as well as through occupying decoding algorithm.

Decoding by Belief Propagation: We represent the sparse by a sparse bipartite graph. In addition to exhilarating the algorithm, the sparse structure reduces the number of loops in the graph and thus supports the convergence of a message passing method that helps in solving Bayesian inference problem. We implement BP in a manner similar to LDPC channel decoding [14], [15]. To decode a length- N signal containing K large coefficients, our BCS-BP decoding algorithm uses measurements and computation. Despite BCS-BP is not guaranteed to mingle but numerical results are quite favourable.

2. RELATED WORK

We use the two-state Gaussian mixture distribution to model the coefficients of approximately sparse signals [17]. This problematic model for the signal in brief captures our prior knowledge about its sparsity. We describe the signal model below.

Let $X = [X(1), X(2), \dots, X(N)]$ be a random vector in $\mathbb{R}^N$, and let us consider the signal $x = [x(1), x(2), \dots, x(N)]$ as an outcome of X . The main observation we deed is that a sparse signal consists of a small number of large coefficients and a large number of small coefficients. Thus each coefficient of the signal can be related with a state variable that can take on two values: for large magnitude coefficients, “high”, and “low”, representing a coefficient of small magnitude. Let the state associated with the i th coefficient be denoted by $q(i)$, where either $q(i) = 1$ (high) or $q(i) = 0$ (low). We view $q(i)$ as an result of $Q(i)$ which represents the state random variable that can take values from $\{0, 1\}$. Let $Q = [Q(1), Q(2), \dots, Q(N)]$ be the state random vector associated with the signal. The actual state configuration $q = [q(1), q(2), \dots, q(N)] \in \{0, 1\}^N$ is one of 2^N possible outcomes of Q . We associate with each possible state of coefficient j a probability density for $X(i)$, resulting in a two state mixture distribution for that coefficient. For the “high” state, we choose a high-variance zero mean Gaussian distribution, and for the “low” state, we choose a low-variance zero-mean Gaussian distribution to model the coefficients.

Thus the distribution of $X(i)$ conditioned on $Q(i)$ is given by

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

Where $\sigma_1 > \sigma_0$. For simplicity, we assume independence between coefficients: the outcome of the state or value of a coefficient does not influence the state or value of another coefficient. Finally, to ensure that we have about K large magnitude coefficients, we choose the probability mass function (pmf) of the state variable $Q(i)$ to be Bernoulli with $P [Q(i) = 1] = S$ and $P [Q(i) = 0] = 1-S$, where $S = K/N$ is the sparsity rate (we assume that the sparsity rate is known).

Fig. 1. The two-state Gaussian mixture model is completely characterized by three parameters: the distribution of the state variable and the variances σ_1^2 and σ_0^2 , of the Gaussian pdf's corresponding to each state.

3. SPARSE CS ENCODING MATRIX

We take sparse matrix to expedite both Compressive Sensing encoding and decoding. Our CS encoding matrices are prevailed by zero entries, with a small number of nonzero in each row and each column. We target on CS-LDPC matrices whose nonzero entries are $\{-1, 1\}$; each measurement involves only sums and differences of a small subset of coefficients of $\mathbf{x}$. Although the coherence between a sparse Φ and Ψ , which is the largest inner product between rows of Φ and Ψ , may be higher than the coherence using a dense Φ matrix, as long as Φ is not too sparse the measurements capture enough information about $\mathbf{x}$ to decode the signal. A Φ matrix can be interpreted as a bipartite graph G , whose edges attach to a coefficient $x(i)$, node to an encoding node $y(j)$, and corresponds to a nonzero entry of Φ . (Fig. 2.)

In the core structure of matrix Φ , we can discuss other restraints to counter the measurement process to the signal model. The constant row weight restraints assured that each row of Φ contains exactly L nonzero entries. The row weight L can be selected based on signal properties such as sparsity, probable noise, and particulars of the decoding process. Another choice is to use a constant column weight restraints, which scrapes the number of nonzero entries in each column of Φ to be a constant R . Although our insistence is on noiseless measurement of approximately sparse signals, we summarily review noisy measurement of a strictly sparse signal, and show that a constant row weight L ensures that the measurements are approximated by two-state mixture Gaussians. To see this, consider a strictly sparse x with sparsity rate S and Gaussian variance σ^2 . We now have $y = \Phi x + z$ where $z \sim (0, \sigma^2)$ is additive white Gaussian noise (AWGN) with variance σ^2 .

Fig. 2. Factor graph depicting the relationship between variable nodes (black) and constraint nodes (white) in CS-BP.

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 $z \sim (0, \sigma_z^2)$ is additive white Gaussian noise (AWGN) with variance σ_z^2 . In our approximately sparse setting, each row of Φ picks up $\approx L(I - S)$ small magnitude coefficients. If $L(I - S)\sigma_0^2 \approx \sigma_z^2$ then the few large coefficients will be obscured by similar noise artifacts.

Matrix Φ relies on the constant assumption that p is sparse in the approved infrequent or sparse basis, i.e., $\Psi = I$. In contrast, if x is infrequent in some other basis Ψ , then complex encoding matrices may be necessary. We defer the discussion of these issues to next Section, but highlighted that in many practical situations our methods can be extended to support the infrequent basis Ψ in a computationally tractable manner.

Information content of Sparsely Encoded Measurements: The sparsity of our CS-LDPC matrix may outturns measurements y that contain few information about the signal x than a dense Gaussian Φ . The following theorem, whose proof appears in the Appendix, verifies that y retains enough information to decode x well. As long as $S = K/N = \Omega((\sigma_0/\sigma_1)^2)$, then $M = O(K \log(N))$ measurements are sufficient.

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

$$M = O\left(\frac{(1 + 2\eta^{-1})(1 + \gamma)}{\mu^2} \left[2K + (N - K) \left(\frac{\sigma_0}{\sigma_1}\right)^2\right] \log(N)\right)$$

The proof of Theorem 1 relies on a result by Wang et al. ([18], Theorem 1). Their proof partitions Φ into M_2 submatrices of M_1 rows each, and estimates each $\hat{x}_i$ as a median of inner products with submatrices. 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:

$$L \approx S^{-1} = \frac{N}{K}, \quad M = O(K \log(N)), \quad \text{and} \\ R = \frac{LM}{N} = O(\log(N)) \quad (4)$$

Noting that each measurement requires $O(L)$ additions and subtractions, and using our rules of thumb for L and M (3.2), the computation required for encoding is $O(LM) = O(N \log(N))$, which is significantly lower than the $O(MN) = O(KN \log(N))$ required for dense Gaussian Φ . In addition to Theorem 1, recent results indicate that there is no loss of optimality with respect to any random matrix distribution when using a sparse CS-LDPC matrix [11]; CS-BP is asymptotically optimal in the large system limit.

4. BCS-BP DECODING OF APPROXIMATELY SPARSE SIGNAL

Decoding relatively sparse random signals can be managed as a Bayesian inference problem. We examine the measurements $y = \Phi x$, where x is a mixture Gaussian signal. Our intention is to estimate x given Φ and y . Because the set of equations $y = \Phi x$ is under-determined, there are infinitely many solutions. All explanations lie along a hyperplane of dimension $N - M$. We locate the solution within this hyperplane which about our prior signal model. Consider the minimum mean-square error (MMSE) and maximum a posteriori (MAP) estimates:

$$\hat{x}_{\text{MMSE}} = \arg \min E \|X - x'\|_2^2 \text{ s.t. } y = \Phi x'$$

$$\hat{x}_{\text{MAP}} = \arg \max f(X = x') \text{ s.t. } y = \Phi x$$

where the expectation is taken over the prior distribution for X . The MMSE estimate can be expressed as the conditional mean, $\hat{x}_{\text{MMSE}} = E[X / Y = y]$, where $Y \in \mathbb{R}^M$ is the random vector that corresponds to the measurements. Although the precise computation of $\hat{x}_{\text{MMSE}}$ may require the evaluation of 2^N terms, a close approximation to the MMSE estimate can be obtained using the (usually small) set of state configuration vectors q with dominant posterior probability [9]. Indeed, exact inference in graphical models is NP-hard 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.

4.1 Decoding Algorithm:

We now employ belief propagation (BP), an efficient method for solving inference problems by iteratively passing messages over graphical models [19]. Although BP has not been proved to assemble, for graphs with few loops it usually offers a better approximation to the solution to the MAP inference problem. BP confides on factor graphs, which enable fast computation of global multivariate functions by exploiting as the global function factors into a product of simpler local functions, each of which depends on a subset of variables [20].

Factor Graph for CS-BP: The factor graph shown in Fig. 4.1 captures the relationship between the states q , the 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 $Q(j)$, coefficient variables $X(j)$, and measurement variables $Y(i)$. The factor graph also has three types of constraint nodes, which encapsulate the dependencies that their neighbours 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: BCS-BP resembles the minimal distributions of all coefficient and state variables in the factor graph, conditioned on the recognized measurements Y , by fleeting messages between variable nodes and constraint nodes. Each message encodes the minimal distributions of a variable related with one of the edges. Given the distributions $Pr(Q(i)/Y=y)$ and $f(X(i)/Y=y)$, one can excerpt MAP and MMSE estimates for each coefficient.

Denote the message sent from a variable node v to one of its neighbors in the bipartite graph, a constraint node c , by $\mu_v \rightarrow c(v)$; a message from c to v is denoted by $\mu_c \rightarrow v(v)$. The message $\mu_v \rightarrow c(v)$ is updated by taking the product of all messages received by v on all other edges. The message $\mu_c \rightarrow v(v)$ is enumerated in a identical manner, but the necessity correlated with c is applied to the product and the result is marginalized. More formally,

$$\mu_v \rightarrow c(v) = \prod_{u \in n(v) \setminus \{c\}} \mu_u \rightarrow v(v) \quad (5)$$

$$\mu_c \rightarrow v(v) = \sum_{\sim\{v\}} \left(\text{con}(n(c)) \prod_{w \in n(c) \setminus \{v\}} \mu_w \rightarrow c(w) \right) \quad (6)$$

Where $n(v)$ and $n(c)$ are sets of neighbors of v and c , respectively, $con(n(c))$ is the constraint on the set of variable nodes $n(c)$, and $\sim \{v\}$ is the set of neighbors of excluding. We interpret these 2 types of message processing as multiplication of beliefs at variable nodes (5) and convolution at constraint nodes (6). 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.

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

$$f(v) = \prod_{u \in n(c) \setminus \{v\}} \mu_{u \rightarrow v}(v) \quad (7)$$

We also need a process to encode beliefs. There are two methods i) one is to sample the relevant pdf's uniformly and then use samples as messages, ii) another 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 trade-offs between modeling flexibility and computational requirements; details appear in Section 4.2.

Protecting Against Loopy Graphs and Message Quantization: Errors: BP assembles to the exact conditional distribution in the optimal situation where the consecutive conditions are met: i) cycle-free factor graph; and ii) messages are processed and propagated without errors. In CS-BP decoding, both conditions are disobeyed. First, the factor graph is loopy- it consists of cycles. Second, message encoding methods introduce errors. These non-awareness may lead BCS-BP to assemble to imprecise conditional distributions, or more critically, lead BCS-BP to diverge [21]. To a few extents these problems can be reduced by employing CS-LDPC matrices, which have a relatively moderate number of loops and together with that correctly designing our message encoding techniques (Sections 4.2). We secure BCS-BP against these non-awareness using message damped belief propagation (MDBP), where messages are weighted averages between old and new estimates. BCS-BP is not assured to converge due to damping, but the arithmetical results demonstrate that its performance is quite satisfactory.

BCS-BP Decoding Algorithm:

1. **Initialization:** Initialize the iteration counter $i = 1$. Set up data structures for factor graph messages $\mu_v \rightarrow c(v)$ and $\mu_c \rightarrow v(v)$. Initialize messages $\mu_c \rightarrow v(v)$ from variable to constraint nodes with the signal prior.
2. **Convolution:** For each measurement $c = 1, 2, \dots, M$, which corresponds to constraint node c , compute $\mu_c \rightarrow v(v)$ via convolution (6) 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 [55] by weighting the new estimates from iteration with estimates from previous iterations.
3. **Multiplication:** For each coefficient $v = 1, 2, \dots, N$, which corresponds to a variable node v , compute $\mu_v \rightarrow c(v)$ via multiplication (5) 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 $v = 1, 2, \dots, N$, compute MMSE or MAP estimates (or alternative statistical characterizations) based on the marginal distribution $f(v)$ (7). Output the requisite statistics.

4.2 Samples of PDF as Messages:

Having discussed main aspects of the BCS-BP decoding algorithm, we now target 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 (5) corresponds to point-wise multiplication of messages; convolution (6) is computed efficiently in the frequency domain.

The main profit of using samples is flexibility to distinct former distributions for the coefficients; for example, mixture Gaussian priors are easily supported. Additionally, both multiplication and convolution are computed efficiently. Sampling has large memory requirements and introduces quantization errors that reduce precision and hamper the convergence of BCS-BP [21]. Sampling also lacks exquisite sampling for accurate 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 (5) and convolution at constraint nodes (6). Outgoing messages are modified,

$$\mu_v \rightarrow c(v) = \frac{\prod_{u \in n\{v\}} \mu_u \rightarrow v(v)}{\mu_c \rightarrow v(v)} \quad \text{and}$$

$$\mu_c \rightarrow v(v) = \sum_{\sim\{v\}} \left(\text{con}(n(c)) \frac{\prod_{w \in n\{c\}} \mu_w \rightarrow c(w)}{\mu_v \rightarrow c(v)} \right) \quad (8)$$

where the denominators are nonzero, because mixture Gaussian pdf's are strictly positive. The modifications (8) reduce computation, because the numerators are computed once and then reused for all messages leaving the node being processed.

Assuming that the column weight 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 requires $O(NRp)$ computation. 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 . 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 [22]. Second, sampling the pdf's with a spacing less than σ_0 , we choose $p = O(\sigma_1 / \sigma_0)$ to support a maximal amplitude on the order of σ_1 . Therefore, our overall computation is $(\sigma_1 / \sigma_0 \log(\sigma_1 / \sigma_0) N \log^2(N))$, which scales as $O(N \log^2(N))$ when σ_0 and σ_1 are constant.

4.3 Properties of BCS-BP Decoding:

There are several properties of BCS-BP decoding some of which are briefly described below. The computational characteristics of the two methods for encoding beliefs about conditional distributions were evaluated in Section 4.2. 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 $O(mN \log(N))$.

Various other properties are briefly explained. There is robustness to noise, because noisy measurements can be assimilated into our model by convolute the noiseless version of the estimated pdf at each encoding node with the pdf of the noise. We have done radical decoding as more measurements will improve the exactness of the estimated posterior probabilities. If we want to estimate the state configuration vector q but not the coefficient values, then less information will be obtained from the measurements. Hence, the number of measurements can be decreased.

5. ARITHMETICAL RESULTS

To demonstrate the adequacy of BCS-BP, we simulated various different settings. In our first setting, we considered decoding problems where $N = 1000$, $S = 0.1$, No. of Repitation (NR) = 50, Iterations (ITR) = 15 and the measurements are noiseless. We used samples of the pdf as messages, where each message consisted of $p = 243$ samples; this choice of provided fast FFT computation. Fig. 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 (top) and of the small coefficients M (bottom); increasing 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, fewer measurements are needed to obtain the same precision. However, there is an optimal $L_{\text{opt}} \approx 2/S_0 = 20$ beyond which any performance gains are marginal. Furthermore, values of $L > L_{\text{opt}}$ give rise to divergence in BCS-BP, even with damping.

Fig. 3. MMSE as a function of the number of measurements M using different matrix row weights L . $N = 1000$, $SNR = 100$, $NR = 50$, $ITR=15$, and noiseless measurements.).

Fig. 4. MMSE as a function of the number of measurements M using different matrix row weights L . $N = 800$, $SNR = 100$, $NR = 20$, $ITR=15$, and noiseless measurements.).

Fig. 5. MMSE as a function of the number of measurements M using different matrix row weights L . $N = 800$, $SNR = 120$, $NR = 20$, $ITR=15$, and noiseless measurements.)

Fig. 6. MMSE as a function of the number of measurements M using different matrix row weights L . $N = 800$, $SNR = 130$, $NR = 20$, $ITR=15$, and noiseless measurements.)

Fig. 7. MMSE as a function of the number of measurements M using different matrix row weights L . $N = 800$, $SNR = 150$, $NR = 20$, $ITR=15$, and noiseless measurements.)

Fig. 8. Original signal x ($N = 1000$, $SNR = 100$, $L = 20$, $M = 400$ and noiseless measurements).

Fig. 9. Decoded signal by CS-BP ($N = 1000$, $SNR = 100$, $L = 20$, $M = 400$ and noiseless measurements).

6. CONCLUSION

Our method developed BCS-BP algorithm which comprises of sparse encoding matrices and a belief propagation decoding algorithm under Bayesian framework. Whereas, we focus on decoding approximately sparse signals, CS-BP can be extended to a signal that are sparse in other bases, is flexible to modifications in the signal model, and can address measurement noise. Even with all crucial profits, BCS-BP is not global as the encoding sparse matrix and decoding method should be modified to use our groundwork to arbitrary bases.

Our work resembles LDPC codes [14], [15]; which use a sparse Bernoulli parity check matrix. A comparison of the MMSE performance of BCS-BP with information theoretically bounds on Compressive Sensing performance has indicated that BCS-BP is not to meet optimal in the large-system limit [11].

With respect to prior work on Bayesian facets of CS [7], [8]; our technique is much faster, which only requires calculations. At the same time, BCS-BP offers remarkable flexibility and should not be considered as merely another fast CS decoding algorithm. However, BCS-BP depends on the sparsity of CS-LDPC matrices, and future work can be considered as the applicability of such matrices in different applications.

REFERENCES

- [1]. Emmanuel J. Candès and Michael B. Wakin, “An introduction to Compressive sampling”, IEEE SIGNAL PROCESSING MAGAZINE.
- [2]. Richard G. Baraniuk, “Compressive Sensing,” IEEE SIGNAL PROCESSING MAGAZINE JULY 2007.
- [3]. E. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” IEEE Trans. Inf. Theory, vol. 52, no. 2, pp. 489–509, Feb. 2006.
- [4]. D. Donoho, “Compressed sensing,” IEEE Trans. Inf. 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. Comput. Math., pp. 295–308, 2005.
- [6]. S. Chen, D. Donoho, and M. Saunders, “Atomic decomposition by basis pursuit,” SIAM J. Sci. Comput., vol. 20, no. 1, pp. 33–61, 1998.
- [7]. S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing,” IEEE Trans. Signal Process., vol. 56, no. 6, pp. 2346–2356, Jun. 2008.
- [8]. M. W. Seeger and H. Nickisch, “Compressed sensing and Bayesian experimental design,” in Proc. 25th Int. Conf. Machine Learning (ICML), 2008, pp. 912–919.
- [9]. P. Schniter, L. C. Potter, and J. Ziniel, “Fast Bayesian matching pursuit: Model uncertainty and parameter estimation for sparse linear models,” Mar. 2009.
- [10]. T. Hastie, R. Tibshirani, and J. H. Friedman, The Elements of Statistical Learning. New York: Springer, Aug. 2001.
- [11]. G. Guo and C.-C. Wang, “Multiuser detection of sparsely spread CDMA, IEEE J. Sel. Areas Commun, vol. 26, no. 3, pp. 421–431, 2008.
- [12]. J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA: Morgan-Kaufmann, 1988.
- [13]. N. Sommer, M. Feder, and O. Shalvi, “Low-density lattice codes,” IEEE Trans. Inf. Theory, vol. 54, no. 4, pp. 1561–1585, 2008.
- [14]. R. G. Gallager, “Low-density parity-check codes,” IEEE Trans. Inf. Theory, vol. 8, no. 1, pp. 21–28, Jan. 1962.
- [15] Anoop Joyti Sahoo, and Rajesh Kumar Tiwari “A Novel Approach for Hiding Secret data in Program Files” International Journal of Information and Computer Security. Volume 8 Issue 1, March 2016,
- [16] Abu Salim, Sachin Tripathi and Rajesh Kumar Tiwari “A secure and timestamp-based communication scheme for cloud environment” Published in International Journal of Electronic Security and Digital Forensics, Volume 6, Issue 4, 319-332.
- [17] Rajesh Kumar Tiwari and G. Sahoo, “A Novel Watermark Scheme for Secure Relational Databases” Information Security Journal: A Global Perspective, Volume 22, Issue 3, July 2013.
- [18]. T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” IEEE Trans. Inf. Theory, vol. 47, pp. 619–637, Feb. 2001.
- [19]. R. Berinde and P. Indyk, “Sparse recovery using sparse random matrices,” MIT, Cambridge, MA, Tech. Rep. MIT-CSAIL-TR-2008-001, 2008.
- [20]. J.-C. Pesquet, H. Krim, and E. Hamman, “Bayesian approach to best basis selection,” in Proc. IEEE 1996 Int. Conf. Acoustics, Speech, Signal Process. (ICASSP), 1996, pp. 2634–2637.
- [21]. W. Wang, M. Garofalakis, and K. Ramchandran, “Distributed sparse random projections for refinable approximation,” in Proc. Inf. Process. Sensor Networks (IPSN), 2007, pp. 331–339.
- [22]. J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Mateo, CA: Morgan-Kaufmann, 1988.
- [23]. F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [24]. E. Sudderth, A. Ihler, W. Freeman, and A. S. Willsky, “Nonparametric belief propagation,” MIT, Cambridge, MA, MIT LIDS Tech. Rep. 2551, Oct. 2002.
- [25]. D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inf. Theory, vol. 45, no. 11, pp. 399–431, Mar. 2005.