Bayesian Compressive Sensing

Shihao Ji, Ya Xue and Lawrence Carin*

EDICS: DSP-RECO

Abstract

The data of interest are assumed to be represented as N-dimensional real vectors, and these vectors are compressible in some linear basis $\boldsymbol{B}$, implying that the signal can be reconstructed accurately using only a small number $M \ll N$ of basis-function coefficients associated with $\boldsymbol{B}$. Compressive sensing is a framework whereby one does not measure one of the aforementioned N-dimensional signals directly, but rather a set of related measurements, with the new measurements a linear combination of the original underlying N-dimensional signal. The number of required compressive-sensing measurements is typically much smaller than N, offering the potential to simplify the sensing system. Let $\boldsymbol{f}$ denote the unknown underlying N-dimensional signal, and $\boldsymbol{g}$ a vector of compressive-sensing measurements, then one may approximate $\boldsymbol{f}$ accurately by utilizing knowledge of the (under-determined) linear relationship between $\boldsymbol{f}$ and $\boldsymbol{g}$, in addition to knowledge of the fact that $\boldsymbol{f}$ is compressible in $\boldsymbol{B}$. In this paper we employ a Bayesian formalism for estimating the underlying signal $\boldsymbol{f}$ based on compressive-sensing measurements $\boldsymbol{g}$. The proposed framework has the following properties: (i) in addition to estimating the underlying signal $\boldsymbol{f}$, "error bars" are also estimated, these giving a measure of confidence in the inverted signal; (ii) using knowledge of the error bars, a principled means is provided for determining when a sufficient number of compressive-sensing measurements have been performed; (iii) this setting lends itself naturally to a framework whereby the compressive-sensing measurements are optimized adaptively and hence not determined randomly; and (iv) the framework accounts for additive noise in the compressive-sensing measurements and provides an estimate of the noise variance. In this paper we present the underlying theory, an associated algorithm, example results, and provide comparisons to other compressive-sensing inversion algorithms in the literature.

Index Terms

Compressive sensing (CS), Sparse Bayesian learning, Relevance vector machine (RVM), Experiment design, Projection optimization.

Contact Information:

Shihao Ji
Department of Electrical and Computer Engineering
Duke University, Box 90291, Durham, NC 27708, USA
Email: shji@ee.duke.edu

Ya Xue
Centice Corporation
215 Southport Dr., Suite 1000, Morrisville, NC 27560, USA
Email: yxue@centice.com

Lawrence Carin*
Department of Electrical and Computer Engineering
Duke University, Box 90291, Durham, NC 27708, USA
Tel: 919-660-5270, Fax: 919-660-5293
Email: lcarin@ee.duke.edu

* Corresponding author.

I. INTRODUCTION

Over the last two decades there have been significant advances in the development of orthonormal bases for compact representation of a wide class of discrete signals. An important example of this is the wavelet transform [1], [2], with which general signals are represented in terms of atomic elements localized in time and frequency, yielding highly compact representations of many natural signals. Let the $N \times N$ matrix $\boldsymbol{B}$ represent a wavelet basis, with basis functions defined by associated columns; a general signal $\boldsymbol{f} \in \mathbb{R}^N$ may be represented as $\boldsymbol{f} = \boldsymbol{B}\boldsymbol{w}$, where $\boldsymbol{w} \in \mathbb{R}^N$ represents the wavelet and scaling function coefficients [1], [2]. For most natural signals $\boldsymbol{f}$, most components of the vector $\boldsymbol{w}$ have negligible amplitude. Therefore, if $\hat{\boldsymbol{w}}$ represents the weights $\boldsymbol{w}$ with the smallest $N-M$ coefficients set to zero, and $\hat{\boldsymbol{f}} = \boldsymbol{B}\hat{\boldsymbol{w}}$, then the relative error $||\boldsymbol{f} - \hat{\boldsymbol{f}}||_2/||\boldsymbol{f}||_2$ is often negligibly small for $M \ll N$. This property has led to the development of state-of-the-art compression algorithms based on wavelet-based transform coding [3], [4].

In conventional applications one first measures the N-dimensional signal $\boldsymbol{f}$, $\boldsymbol{f}$ is then compressed (often using a wavelet-based transform encoding scheme), and the compressed set of basis-function coefficients $\boldsymbol{w}$ are stored in binary [3], [4]. This invites the following question: If the underlying signal is ultimately compressible, is it possible to perform a compact ("compressive") set of measurements directly, thereby offering the potential to simplify the sensing system (reduce the number of required measurements)? This question has recently been answered in the affirmative, introducing the field of compressive sensing (CS) [5], [6].

In its earliest form the relationship between the underlying signal $\boldsymbol{f}$ and the CS measurements $\boldsymbol{g}$ has been constituted through random projections [5], [6]. Specifically, assume that the signal $\boldsymbol{f}$ is compressible in some basis $\boldsymbol{B}$ (not necessarily a wavelet basis), the kth CS measurement g_k (kth component of $\boldsymbol{g}$) is constituted by projecting $\boldsymbol{f}$ onto a "random" basis that is constituted with "random" linear combination of the basis functions in $\boldsymbol{B}$, i.e., $g_k = \boldsymbol{f}^T(\boldsymbol{B}\boldsymbol{r}_k)$, where $\boldsymbol{r}_k \in \mathbb{R}^N$ is a column vector with each element an i.i.d. draw of a random variable, with arbitrary alphabet (e.g., real or binary) [7].

Based on the above discussion, the CS measurements may be represented as $\boldsymbol{g} = \boldsymbol{\Phi}\boldsymbol{B}^T\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{w}$, where $\boldsymbol{\Phi} = [\boldsymbol{r}_1 \ldots \boldsymbol{r}_K]^T$ is an $K \times N$ matrix, assuming K random measurements are made. Since typically $K < N$ we have fewer measurements than degrees of freedom for the signal $\boldsymbol{f}$. Therefore, inversion for the weights $\boldsymbol{w}$ (and hence $\boldsymbol{f}$) is ill-posed. However, if one exploits the fact that $\boldsymbol{w}$ is sparse with respect to a known orthonormal basis $\boldsymbol{B}$, then one may approximate $\boldsymbol{w}$ accurately via an ℓ_1-regularized

formulation [5], [6]

$$\tilde{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \{||\boldsymbol{g} - \boldsymbol{\Phi}\boldsymbol{w}||_2^2 + \rho||\boldsymbol{w}||_1\}, \quad (1)$$

where the scalar ρ controls the relative importance applied to the Euclidian error and the sparseness term (the first and second expressions, respectively, inside the brackets in (1)). This basic framework has been the starting point for several recent CS inversion algorithms, including linear programming [8] and greedy algorithms [9], [10] for estimation of the weights $\boldsymbol{w}$.

In this paper we consider the inversion of compressive measurements from a Bayesian perspective. Specifically, from this standpoint we have a prior belief that $\boldsymbol{w}$ should be sparse in the basis $\boldsymbol{B}$, data $\boldsymbol{g}$ are observed from compressive measurements, and the objective is to provide a posterior belief (density function) for the values of the weights $\boldsymbol{w}$. Hence, rather than providing a point (single) estimate for the weights $\boldsymbol{w}$, a full posterior density function is provided, which yields "error bars" on the estimated $\boldsymbol{f}$. These error bars may be used to give a sense of confidence in the approximation to $\boldsymbol{f}$, and they may also be used to guide the optimal design of additional CS measurements, implemented with the goal of reducing the uncertainty in $\boldsymbol{f}$. We also demonstrate that the conventional point estimate represented by (1) is a special case of the Bayesian formalism; specifically, (1) is a maximum *a posterior* (MAP) estimate of the weights $\boldsymbol{w}$.

In addition to presenting the Bayesian form of CS, as discussed above, we also present an efficient algorithm for its implementation. This algorithm is compared with existing CS inversion algorithms [8], [10], for canonical data considered in the literature.

The remainder of the paper is organized as follows. In Sec. II we consider the CS-inversion problem from a Bayesian perspective, and make connections with what has been done previously for this problem. The analysis is then generalized in Sec. III, yielding a framework that lends itself to efficient computation of an approximation to a posterior density function for $\boldsymbol{w}$ (and hence for $\boldsymbol{f}$). In Sec. IV we examine how this framework allows adaptive CS, whereby the aforementioned projects $\boldsymbol{r}_k$ are selected to optimize a (myopic) information measure. Example results on canonical data are presented in Sec. V, with comparisons to other CS inversion algorithms currently in the literature. Conclusions and future work are discussed in Sec. VI.

II. Compressive-Sensing Inversion From Bayesian Viewpoint

A. Compressive Sensing as Linear Regression

It was assumed at the start that $\boldsymbol{f}$ is compressible in the basis $\boldsymbol{B}$, and the CS measurements $\boldsymbol{g}$ are performed in the form of random projections, $\boldsymbol{g} = \boldsymbol{\Phi}\boldsymbol{B}^T\boldsymbol{f} = \boldsymbol{\Phi}\boldsymbol{w}$. Therefore, let $\boldsymbol{w}_s$ represent an N-

dimensional vector that is identical to the vector $\boldsymbol{w}$ for the M elements in $\boldsymbol{w}$ with largest magnitude; the remaining $N-M$ elements in $\boldsymbol{w}_s$ are set to zero. Similarly, we introduce a vector $\boldsymbol{w}_e$ that is identical to $\boldsymbol{w}$ for the smallest $N-M$ elements in $\boldsymbol{w}$, with all remaining elements of $\boldsymbol{w}_e$ set to zero. We therefore have $\boldsymbol{w} = \boldsymbol{w}_s + \boldsymbol{w}_e$, and

$$\boldsymbol{g} = \boldsymbol{\Phi}\boldsymbol{w} = \boldsymbol{\Phi}\boldsymbol{w}_s + \boldsymbol{\Phi}\boldsymbol{w}_e = \boldsymbol{\Phi}\boldsymbol{w}_s + \boldsymbol{n}_e, \tag{2}$$

where $\boldsymbol{n}_e = \boldsymbol{\Phi}\boldsymbol{w}_e$. Since it was assumed at the start that $\boldsymbol{\Phi}$ is constituted through random samples, the components of $\boldsymbol{n}_e$ may be approximated as a zero-mean Gaussian noise as a consequence of Central-Limit Theorem [11] for large $N-M$. We also note that the CS measurements may be noisy, with the measurement noise, denoted by $\boldsymbol{n}_m$, represented by a zero-mean Gaussian distribution, and therefore

$$\boldsymbol{g} = \boldsymbol{\Phi}\boldsymbol{w}_s + \boldsymbol{n}_e + \boldsymbol{n}_m = \boldsymbol{\Phi}\boldsymbol{w}_s + \boldsymbol{n}, \tag{3}$$

where the components of $\boldsymbol{n}$ are approximated as a zero-mean Gaussian noise with unknown variance σ^2. We henceforth simply utilize (3) as a model, motivated for the reasons discussed above, and desirable from the standpoint of analysis. In practice not all of the assumptions made in deriving (3) will necessarily be valid, but henceforth we simply use (3) as a starting point.

The above analysis has converted the CS problem of inverting for the sparse weights $\boldsymbol{w}_s$ into a linear-regression problem with a constraint (prior) that $\boldsymbol{w}_s$ is sparse, or more relevantly, sparse Bayesian regression [12], [13]. Using an analysis based on (3), we first demonstrate that the conventional CS analysis reflected in (1) is in fact a simplified form of Bayesian inversion. Assuming knowledge of $\boldsymbol{\Phi}$, the quantities to be estimated based on the CS measurements $\boldsymbol{g}$ are the sparse weights $\boldsymbol{w}_s$ and the noise variance σ^2. In a Bayesian analysis we seek a full posterior density function for $\boldsymbol{w}_s$ and σ^2.

B. Sparseness Prior and MAP approximation

In a Bayesian formulation our understanding of the fact that $\boldsymbol{w}_s$ is sparse is formalized by placing a sparseness-promoting prior on $\boldsymbol{w}_s$. A widely used sparseness prior is the Laplace density function [14]:

$$p(\boldsymbol{w}|\lambda) = (\lambda/2)^N \exp(-\lambda \sum_{i=1}^{N} |w_i|), \tag{4}$$

where in (4) and henceforth we drop the subscript s on $\boldsymbol{w}$, recognizing that we are always interested in a sparse solution for the weights (w_i is the ith component of $\boldsymbol{w}$). We will also place a prior on the inverse variance, or "precision", of the noise $\boldsymbol{n}$, i.e., $p(\alpha_0|c,d) = \Gamma(\alpha_0|c,d)$, where $\alpha_0 = 1/\sigma^2$ and c and d are hyperparameters of the Gamma prior [14]; the reason for choosing a Gamma prior on α_0 is discussed in Sec. III-A.

Given the CS measurements $\boldsymbol{g}$, and assuming the model in (3), the likelihood function for the weights $\boldsymbol{w}$ and precision α_0 is

$$p(\boldsymbol{g}|\boldsymbol{w},\alpha_0) = \frac{1}{(2\pi/\alpha_0)^{K/2}} \exp\{-\frac{\alpha_0}{2}||\boldsymbol{g}-\boldsymbol{\Phi}\boldsymbol{w}||_2^2\}, \tag{5}$$

and hence the posterior density function for $\boldsymbol{w}$ and α_0, given hyperparameters c, d and λ on the aforementioned priors, satisfies

$$p(\boldsymbol{w},\alpha_0|\boldsymbol{g},c,d,\lambda) \propto \frac{1}{(2\pi/\alpha_0)^{K/2}} \exp\{-\frac{\alpha_0}{2}||\boldsymbol{g}-\boldsymbol{\Phi}\boldsymbol{w}||_2^2\}(\lambda/2)^N \exp(-\lambda\sum_{i=1}^{N}|w_i|)\Gamma(\alpha_0|c,d), \tag{6}$$

where the normalization constant in the denominator of $p(\boldsymbol{w},\alpha_0|\boldsymbol{g},c,d,\lambda)$ cannot be computed analytically. If we are only interested in determining a maximum *a posteriori* (MAP) estimate for $\boldsymbol{w}$, we consider the log posterior:

$$\log p(\boldsymbol{w},\alpha_0|\boldsymbol{g},c,d,\lambda) = -\frac{\alpha_0}{2}||\boldsymbol{g}-\boldsymbol{\Phi}\boldsymbol{w}||_2^2 - \lambda\sum_{i=1}^{N}|w_i| + const, \tag{7}$$

where $const$ is a constant with respect to $\boldsymbol{w}$. We therefore observe that maximizing the log posterior with respect to $\boldsymbol{w}$ is equivalent to the ℓ_1-regularized formulation in (1), where the value of ρ in (1) is driven by the relative weights of α_0 and λ.

III. Estimate of Full Posterior for Sparse Weights

A. Hierarchical Sparseness Prior

The above discussion demonstrated that conventional CS inversion for the weights $\boldsymbol{w}$ corresponds to a MAP approximation to a Bayesian linear-regression analysis, with a Laplace sparseness prior on $\boldsymbol{w}$. This then raises the question of whether the Bayesian analysis may be carried further, to realize an estimate of the full posterior on $\boldsymbol{w}$ and α_0. The use of a Laplace sparseness prior makes this difficult, due to the complexity of evaluating the denominator of the posterior; this is because the Laplace prior is not conjugate [14] to the likelihood in (6). However, to mitigate this problem, we consider a hierarchical sparseness prior, which lends itself to a relatively simple graphical model, for which the posterior can indeed be approximated accurately in an efficient algorithm.

Rather than imposing a Laplace prior on $\boldsymbol{w}$, we develop a hierarchical prior [12], [15] that has similar properties but that allows convenient conjugate-exponential analysis. Specifically, we introduce the prior

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N}\mathcal{N}(w_i|0,\alpha_i^{-1}), \tag{8}$$

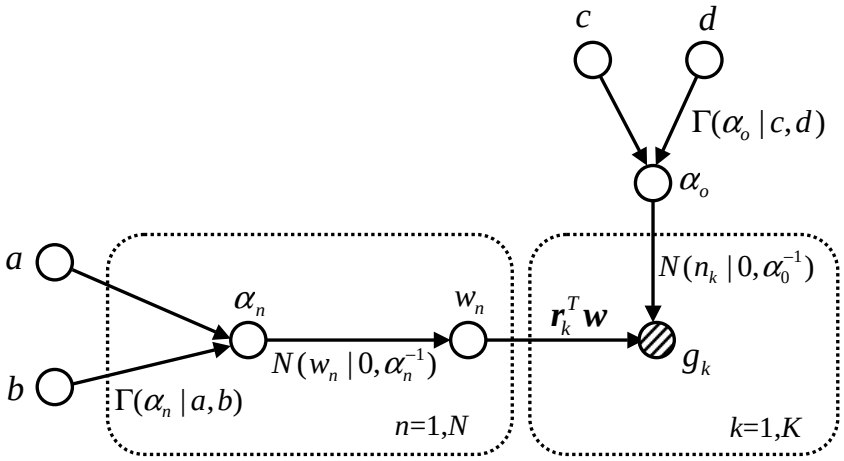

Fig. 1. Graphical model of the Bayesian CS formulation.

where $\mathcal{N}(w_i|0,\alpha_i^{-1})$ is a zero-mean Gaussian density function with precision (inverse-variance) α_i. We place the following prior on $\boldsymbol{\alpha}$

$$p(\boldsymbol{\alpha}|a,b) = \prod_{i=1}^{N} \Gamma(\alpha_i|a,b). \tag{9}$$

The overall prior on $\boldsymbol{w}$ is evaluated as

$$p(\boldsymbol{w}|a,b) = \prod_{i=1}^{N} \int_0^\infty \mathcal{N}(w_i|0,\alpha_i^{-1})\Gamma(\alpha_i|a,b)d\alpha_i. \tag{10}$$

Density function $\Gamma(\alpha_i|a,b)$ is the conjugate prior for α_i, when w_i plays the role of observed data and $\mathcal{N}(w_i|0,\alpha_i^{-1})$ is a likelihood function; consequently the integral $\int_0^\infty \mathcal{N}(w_i|0,\alpha_i^{-1})\Gamma(\alpha_i|a,b)d\alpha_i$ can be evaluated analytically, and it corresponds to the student-t distribution [14]. With appropriate choice of the hyperparameters a and b, the student-t distribution is strongly peaked about $w_i = 0$, and therefore the prior in (10) favors most w_i being zero (i.e., it is a sparseness prior).

While (10) is a sparseness prior, it is more complicated in form than the Laplace prior in (4), and therefore it is not obvious as to why (10) is preferable. To see the advantage of (10), consider the graphical structure of the model as reflected in Fig. 1, for generation of the observed data $\boldsymbol{g}$. Following consecutive blocks in Fig. 1 (following the direction of the arrows), let p_k represent the parameter associated with block k, and p_{k+1} represents the next parameter in the sequence. For all steps in Fig. 1, the density function for p_k is the conjugate-exponential prior for the likelihood defined in terms of the density function for p_{k+1}, assuming that all parameters except p_k are held constant (i.e., all parameters other than p_k temporarily play the role of fixed data). Moreover, under these conditions the density function for p_k may be updated analytically, corresponding to an update of the hyperparameters of the density function

for p_k. This structural form is very convenient for implementing iterative algorithms for evaluation of the posterior density function for $\boldsymbol{w}$ and α_0. For example, one may conveniently implement a Markov chain Monte Carlo (MCMC) [16] or, more efficiently and approximately, a variational Bayesian (VB) analysis [15], [17]. While the VB analysis is efficient relative to MCMC, and it has been found to yield accurate estimates of the posterior in many problems [15], we here consider a type-II maximum-likelihood (ML) analysis, with the objective of achieving highly efficient computations while still preserving accurate results for related problems [12].

B. Iterative Analysis

As shown by Tipping [12], in the context of the relevance vector machine (RVM), if $\boldsymbol{g}$, $\boldsymbol{\alpha}$ and $\alpha_0 = 1/\sigma^2$ are known, then the posterior for $\boldsymbol{w}$ can be expressed analytically as a multivariate normal distribution with mean and covariance:

$$\boldsymbol{\mu} = \alpha_0 \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{g}, \tag{11}$$

$$\boldsymbol{\Sigma} = (\boldsymbol{A} + \alpha_0 \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1}, \tag{12}$$

where $\boldsymbol{A} = diag(\alpha_1, \alpha_2, \ldots, \alpha_N)$. Further, the marginal likelihood for $\boldsymbol{\alpha}$ and α_0, or equivalently, its logarithm $\mathcal{L}(\boldsymbol{\alpha}, \alpha_0)$ may be expressed analytically by integrating out the weights $\boldsymbol{w}$, to yield

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \alpha_0) &= \log p(\boldsymbol{g}|\boldsymbol{\alpha}, \alpha_0) = \log \int p(\boldsymbol{g}|\boldsymbol{w}, \alpha_0) p(\boldsymbol{w}|\boldsymbol{\alpha}) d\boldsymbol{w} \\ &= -\frac{1}{2} \left[K \log 2\pi + \log |\boldsymbol{C}| + \boldsymbol{g}^T \boldsymbol{C}^{-1} \boldsymbol{g} \right], \end{aligned} \tag{13}$$

with

$$\boldsymbol{C} = \sigma^2 \boldsymbol{I} + \boldsymbol{\Phi} \boldsymbol{A}^{-1} \boldsymbol{\Phi}^T. \tag{14}$$

Then a type-II ML solution employs the point estimates for $\boldsymbol{\alpha}$ and α_0 that maximize (13). This can be implemented readily via the EM algorithm or direct differentiation [12], to yield:

$$\alpha_i^{new} = \gamma_i / \mu_i^2, \qquad i \in \{1, 2, ..., N\}, \tag{15}$$

with $\gamma_i \triangleq 1 - \alpha_i \Sigma_{ii}$, where Σ_{ii} is the ith diagonal element from $\boldsymbol{\Sigma}$ in (12), and

$$1/\alpha_0^{new} = \frac{||\boldsymbol{g} - \boldsymbol{\Phi} \boldsymbol{w}||_2^2}{N - \sum_i \gamma_i}. \tag{16}$$

Note that $\boldsymbol{\alpha}^{new}$ and α_0^{new} are a function of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, while $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are a function of $\boldsymbol{\alpha}$ and α_0; this suggests an iterative algorithm, where one iterates between (11)-(12) and (15)-(16), and in this process α_i becomes very large for those w_i that have insignificant amplitudes for representation of $\boldsymbol{g} = \boldsymbol{\Phi} \boldsymbol{w}$.

Only a relatively small set of w_i for which the corresponding α_i remains relatively small contribute representation of $\boldsymbol{g}$, and the level of sparseness (size of M) is determined automatically. Through the estimated α_0 one also realizes an estimate of the variance associated with $\boldsymbol{n}$ in (3). It is also important to note that, as a result of the type-II ML solution, the final iterative algorithm is independent of the settings a, b, c and d on the Gamma hyperprior, and therefore there are no free parameters in the algorithm.

While it is useful to have a measure of uncertainty in the weights $\boldsymbol{w}$, the quantity of most interest is the signal $\boldsymbol{f} = \boldsymbol{B}\boldsymbol{w}$. Since $\boldsymbol{w}$ is drawn from a multivariate Gaussian distribution with mean and covariance defined in (11)-(12), then $\boldsymbol{f}$ is also drawn from a multivariate Gaussian distribution, with mean and covariance

$$E(\boldsymbol{f}) = \boldsymbol{B}\boldsymbol{\mu}, \tag{17}$$

$$Cov(\boldsymbol{f}) = \boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T. \tag{18}$$

The diagonal elements of the covariance matrix in (18) provide "error bars" on the accuracy of the inversion of $\boldsymbol{f}$, as represented in terms of its mean.

While the algorithm described above has been demonstrated to yield highly accurate sparse linear-regression representation [12], we note the following practical limitation. When evaluating (12) one must invert matrices of size $N \times N$, which has order N^3 complexity, thereby making this approach relatively slow for data $\boldsymbol{f}$ of large dimension N. This motivates development of a computationally efficient approach that is related to such CS algorithms as OMP [9] and StOMP [10]. However, while these algorithms yield a single point estimate for $\boldsymbol{f}$, the Bayesian analysis considered here also yields the error bars defined in (18).

C. Fast Bayesian CS algorithm

Although the hyperparameters $\boldsymbol{\alpha}$ and α_0 can be optimized with respect to (13) via the EM algorithm or direct differentiation [12], a more efficient implementation has been derived by analyzing the properties of the marginal likelihood function. This enables a principled and efficient sequential addition and deletion of candidate basis function (columns of $\boldsymbol{\Phi}$) to monotonically maximize the marginal likelihood. In the following, we briefly review some of the key properties of this fast algorithm; for more details one may refer to [18], [19].

Considering the dependence of $\mathcal{L}(\boldsymbol{\alpha}, \alpha_0)$ on a single hyperparameter α_i, $i \in \{1 \cdots N\}$, we can

decompose $\boldsymbol{C}$ in (14) as

$$\begin{aligned}\boldsymbol{C} &= \sigma^2\boldsymbol{I} + \boldsymbol{\Phi}\boldsymbol{A}^{-1}\boldsymbol{\Phi} = \sigma^2\boldsymbol{I} + \sum_{m\neq i}\alpha_m^{-1}\boldsymbol{\phi}_m\boldsymbol{\phi}_m^T + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T \\ &= \boldsymbol{C}_{-i} + \alpha_i^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T, \end{aligned} \tag{19}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_N]$ (note that $\boldsymbol{\Phi}$ has been expressed previously as $\boldsymbol{\Phi} = [\boldsymbol{r}_1 \cdots \boldsymbol{r}_K]^T$), and $\boldsymbol{C}_{-i}$ is $\boldsymbol{C}$ with the contribution of basis function $\boldsymbol{\phi}_i$ removed. Matrix determinant and inverse identities may be used to express

$$|\boldsymbol{C}| = |\boldsymbol{C}_{-i}|\,|1 + \alpha_i^{-1}\boldsymbol{\phi}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{\phi}_i|, \tag{20}$$

$$\boldsymbol{C}^{-1} = \boldsymbol{C}_{-i}^{-1} - \frac{\boldsymbol{C}_{-i}^{-1}\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T\boldsymbol{C}_{-i}^{-1}}{\alpha_i + \boldsymbol{\phi}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{\phi}_i}. \tag{21}$$

From this, we can write $\mathcal{L}(\boldsymbol{\alpha}, \alpha_0)$ as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\alpha}, \alpha_0) &= \mathcal{L}(\boldsymbol{\alpha}_{-i}, \alpha_0) + \frac{1}{2}\left[\log\alpha_i - \log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}\right] \\ &= \mathcal{L}(\boldsymbol{\alpha}_{-i}, \alpha_0) + \ell(\alpha_i, \alpha_0), \end{aligned} \tag{22}$$

where $\boldsymbol{\alpha}_{-i}$ is the same as $\boldsymbol{\alpha}$ except α_i is removed, and

$$s_i \triangleq \boldsymbol{\phi}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{\phi}_i, \quad \text{and} \quad q_i \triangleq \boldsymbol{\phi}_i^T\boldsymbol{C}_{-i}^{-1}\boldsymbol{g}. \tag{23}$$

Analysis of $\ell(\alpha_i, \alpha_0)$ [18] shows that $\mathcal{L}(\boldsymbol{\alpha}, \alpha_0)$ has a unique maximum with respect to α_i:

$$\alpha_i = \frac{s_i^2}{q_i^2 - s_i}, \qquad \text{if } q_i^2 > s_i, \tag{24}$$

$$\alpha_i = \infty, \qquad \text{if } q_i^2 \leq s_i. \tag{25}$$

Recall that setting $\alpha_i = \infty$ is equivalent to $w_i = 0$, and hence removing $\boldsymbol{\phi}_i$ from the representation; hence, (24)-(25) controls the addition and deletion of particular $\boldsymbol{\phi}_i$ from the signal representation. If we perform these operations sequentially for varying i, we realize an efficient learning algorithm.

It is relatively straightforward to compute s_i and q_i for all the basis vector $\boldsymbol{\phi}_i$, including those not currently utilized in the model (i.e., for which $\alpha_i = \infty$). In practice, it is easier to maintain and update

$$S_m = \boldsymbol{\phi}_m^T\boldsymbol{C}^{-1}\boldsymbol{\phi}_m, \quad Q_m = \boldsymbol{\phi}_m^T\boldsymbol{C}^{-1}\boldsymbol{g}, \tag{26}$$

and from these it follows simply:

$$s_m = \frac{\alpha_m S_m}{\alpha_m - S_m}, \qquad q_m = \frac{\alpha_m Q_m}{\alpha_m - S_m}. \tag{27}$$

Note that when $\alpha_m = \infty$, $s_m = S_m$ and $q_m = Q_m$. In practice, it is convenient to utilize the Woodbury identity to obtain the quantities of interest:

$$S_m = \alpha_0 \boldsymbol{\phi}_m^T \boldsymbol{\phi}_m - \alpha_0^2 \boldsymbol{\phi}_m^T \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{\phi}_m, \tag{28}$$

$$Q_m = \alpha_0 \boldsymbol{\phi}_m^T \boldsymbol{g} - \alpha_0^2 \boldsymbol{\phi}_m^T \boldsymbol{\Phi} \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{g}. \tag{29}$$

Here quantities $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}$ contain only those basis vectors that are currently included in the model, and the computation thus scales in the cube of that measure, which is typically only a very small fraction of N. Furthermore, these quantities can all be calculated via "update" formulae with reduced computation (e.g., $\boldsymbol{\Sigma}$ can be computed recursively without the inverse operation; see [19] for more details.).

Compared with the iterative algorithm presented in Sec. III-B, the fast algorithm summarized above operates in a constructive manner, i.e., the initial solution $\boldsymbol{\alpha}$ is notionally set to infinity, indicating an empty model, and then basis functions $\boldsymbol{\phi}_i$ are sequentially added into or deleted from the model to maximize the marginal likelihood. In addition, the inverse operation that is required to compute $\boldsymbol{\Sigma}$ in (12) now is replaced by a more efficiently with a recursive implementation. Unlike other related CS algorithms (e.g., OMP [9] and StOMP [10]), the fast algorithm has the operation of deleting a basis function (i.e., setting $\alpha_i = \infty$). This deletion operation is the likely explanation for the improvement in sparsity of this algorithm demonstrated in the experiments (see Sec. V).

IV. Adaptive Compressive Sensing

A. Selecting Projections to Reduce Signal Uncertainty

In the original CS construction, the projections represented by $\boldsymbol{\Phi} = [\boldsymbol{r}_1 \dots \boldsymbol{r}_K]^T$ were constituted via i.i.d. realizations of an underlying random variable [7]. In addition, previous CS algorithms [6], [9], [10] focused on estimating $\boldsymbol{w}$ have employed a point estimate like that in (1); such approaches do not provide a measure of uncertainty in $\boldsymbol{f}$, and therefore adaptive design of $\boldsymbol{\Phi}$ was previously not feasible. The Bayesian CS (BCS) algorithm introduced in Sec. III-C allows efficient computation of $\boldsymbol{f}$ and associated error bars, and therefore one may consider the possibility of adaptively selecting projections $\boldsymbol{r}_k$ with the goal of reducing uncertainty. Such a framework has been previously studied in the machine-learning community under the name of experiment design or active learning [20]–[22]. Further, the error bars also give a way to determine how many measurements are enough for faithful CS reconstruction, i.e., when the change in the uncertainty is not significant, it may be assumed that one is simply reconstructing the noise $\boldsymbol{n}$ in (3), and therefore the adaptive sensing may be stopped.

As discussed above, the estimated posterior on the signal $\boldsymbol{f}$ is a multivariate Gaussian distribution, with mean $E(\boldsymbol{f}) = \boldsymbol{B}\boldsymbol{\mu}$ and covariance $Cov(\boldsymbol{f}) = \boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T$. The differential entropy [23] for $\boldsymbol{f}$ therefore satisfies:

$$\begin{aligned} h(\boldsymbol{f}) &= \frac{1}{2}\log|\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T| + c = \frac{1}{2}\log|\boldsymbol{\Sigma}| + c \\ &= -\frac{1}{2}\log|\boldsymbol{A} + \alpha_0\boldsymbol{\Phi}^T\boldsymbol{\Phi}| + c, \end{aligned} \tag{30}$$

where c is a constant, independent of $\boldsymbol{\Phi}$. Recall that $\boldsymbol{A} = diag(\alpha_1, \alpha_2, \ldots, \alpha_N)$, and therefore the dependence of the differential entropy on the observed CS measurements $\boldsymbol{g}$ is defined by the point estimates of $\boldsymbol{\alpha}$ and α_0 (from the type-II ML estimates discussed in Sec. III).

We may now ask which new projection $\boldsymbol{r}_{K+1}$ would be optimal for minimizing the differential entropy in (30). Toward this end, we augment $\boldsymbol{\Phi}$ by adding a $(K+1)$th row represented by $\boldsymbol{r}_{K+1}^T$. If we let $h_{new}(\boldsymbol{f})$ represent the new differential entropy as a consequence of adding this new projection measurement, via the matrix determinant identity we have

$$h_{new}(\boldsymbol{f}) = h(\boldsymbol{f}) - \frac{1}{2}\log\left[1 + \alpha_0\boldsymbol{r}_{K+1}^T\boldsymbol{\Sigma}\,\boldsymbol{r}_{K+1}\right], \tag{31}$$

where α_0 and $\boldsymbol{\Sigma}$ are based on estimates found using the previous K random projections. Then the next projection $\boldsymbol{r}_{K+1}$ should be designed to maximize the variance of the *expected* measurement g_{K+1} since

$$\boldsymbol{r}_{K+1}^T\boldsymbol{\Sigma}\,\boldsymbol{r}_{K+1} = \boldsymbol{r}_{K+1}^T Cov(\boldsymbol{w})\,\boldsymbol{r}_{K+1} = Var(g_{K+1}). \tag{32}$$

In other words, the next projection $\boldsymbol{r}_{K+1}$ should be selected to constitute the measurement g_{K+1} for which the data is most uncertain (and hence access to the associated measurement would be most informative).

There are multiple ways this may be utilized in practice. If it is possible to design new projections $\boldsymbol{r}_{K+1}$ adaptively "on the fly", then one might perform an eigen-decomposition of the matrix $\boldsymbol{\Sigma}$, and select for representation of $\boldsymbol{r}_{K+1}$ the eigenvector with largest eigenvalue. Alternatively, if from a hardware standpoint such flexibility in design of $\boldsymbol{r}_{K+1}$ is not feasible, then one might *a priori* design a library $\boldsymbol{L}$ of possible next projections, with $\boldsymbol{r}_{K+1}$ selected from $\boldsymbol{L}$ with the goal of maximizing (32). In the example results in Sec. V, we select the next projection $\boldsymbol{r}_{K+1}$ as the eigenvector of $\boldsymbol{\Sigma}$ that has the largest eigenvalue, but design of an *a priori* library $\boldsymbol{L}$ may be more useful in practice; the design of such a library is an important direction for future work.

An additional issue needs to be clarified if the eigenvector of $\boldsymbol{\Sigma}$ is used as the next projection $\boldsymbol{r}_{K+1}$. Because of the sparse Bayesian solution, $\boldsymbol{\Sigma}$ only employs elements corresponding to the associated nonzero components of $\boldsymbol{w}$ found based on the fast algorithm (i.e., $\boldsymbol{\Sigma}$ is reduced in general to a small

matrix). So when constructing the next projection based on the eigenvector, some entries of $\boldsymbol{r}_{K+1}$ will be empty. If we impute all those empty entries with zeros, we are under the risk of being wrong. The initial estimate of $\boldsymbol{w}$ can be inaccurate; if we impute with zeros, the estimate will be always biased and has no chance to be corrected since the corresponding contributions from underlying true $\boldsymbol{w}$ are always ignored. To mitigate this problem, we impute those empty entries with random samples drawn i.i.d. from a normal distribution $\mathcal{N}(0,1)$. After the imputation, we re-scale the magnitude of the imputed entries to 0.01. In this way, we utilize the optimized projection, and at the same time allow some contributions from the empty entries. Overall, the final projection $\boldsymbol{r}_{K+1}$ has the magnitude $||\boldsymbol{r}_{K+1}||_2 = 1.01$.

B. Approximate Adaptive CS

The error bars on the estimate of $\boldsymbol{f}$ play a critical role in implementing the above adaptive CS scheme, with these a direct product from the Bayesian analysis. Since there are established CS algorithms based on a point estimate of $\boldsymbol{w}$, one may ask whether these algorithms may be modified, utilizing insights from the Bayesian analysis. The advantage of such an approach is that, if possible, one would access some of the advantages of the Bayesian analysis, in an approximate sense, while being able to retain the advantages of existing CS algorithms.

The adaptive algorithm in (31) rely on computation of the covariance matrix $\boldsymbol{\Sigma} = (\boldsymbol{A} + \alpha_0 \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1}$; since $\boldsymbol{\Phi}$ is assumed known, this indicates that what is needed are estimates for α_0 and $\boldsymbol{\alpha}$, the latter required for the diagonal matrix $\boldsymbol{A}$. From (16) we have $\sigma^2 = 1/\alpha_0 = ||\boldsymbol{g} - \boldsymbol{\Phi}\boldsymbol{w}||_2^2/(N - \sum_i \gamma_i)$, where the denominator $N - \sum_i \gamma_i$ may be viewed as an estimate for the number of components of the weight vector $\boldsymbol{w}$ that have negligible amplitude. Consequently, assume that an algorithm such as OMP [9] or StOMP [10] is used to yield a point estimate of the weights $\boldsymbol{w}$, denoted $\boldsymbol{w}_p$, and assume that there are M_0 non-zero elements in $\boldsymbol{w}_p$; then one may approximate the "noise" variance as $\sigma^2 = ||\boldsymbol{g} - \boldsymbol{\Phi}\boldsymbol{w}_p||_2^2/(N - M_0)$.

Concerning the diagonal matrix $\boldsymbol{A}$, it may be viewed as a regularization of the matrix $(\alpha_0 \boldsymbol{\Phi}^T \boldsymbol{\Phi})$, to assure that the matrix inversion is well posed. While the Bayesian analysis in Sec. III indicates that the loading represented by $\boldsymbol{A}$ should be non-uniform, we may simply make $\boldsymbol{A}$ diagonalized uniformly, with value corresponding to a small fraction of the average value of the diagonal elements of $(\alpha_0 \boldsymbol{\Phi}^T \boldsymbol{\Phi})$. In Sec. V, when presenting example results, we make comparisons between the rigorous implementation discussed in Sec. IV-A and the approximate scheme discussed here (as applied to the BCS algorithm). However, similar modifications may be made to other related algorithms, such as OMP and StOMP.

V. Example Results

We test the performance of BCS for signal/image reconstruction on several example problems considered widely in the literature, with comparisons made to Basis Pursuit (BP) [6] and StOMP [10]. While BP is a relatively slow algorithm that involves linear programming, StOMP may be one of the state-of-the-art fast CS algorithms. To make the speed comparison among different algorithms as fair as possible, all computations presented here were performed using Matlab run on a 3.4GHz Pentium machine.

A. BCS and Projection Optimization

In the first example we consider a length $N = 512$ signal that contains $M = 20$ spikes created by choosing 20 locations at random and then putting ± 1 at these points (Fig. 2(a)). The projection matrix $\mathbf{\Phi}$ is constructed by first creating a $K \times N$ matrix with i.i.d. draws of a Gaussian distribution $\mathcal{N}(0, 1)$, and then the rows of $\mathbf{\Phi}$ are normalized to unit amplitude. To simulate measurement noise, zero-mean Gaussian noise with standard deviation $\sigma_m = 0.005$ is added to each of the K measurements that define the data $\boldsymbol{g}$. In the experiment $K = 100$, and the reconstructions are implemented by BP and BCS. For the BP implementation, we used the ℓ_1-*magic* package available online at `http://www.acm.caltech.edu/l1magic/`.

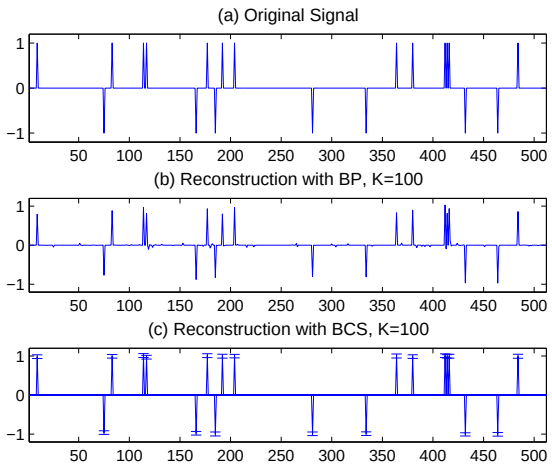

Fig. 2. Reconstruction of *Spikes* for $N = 512$, $M = 20$, $K = 100$. (a) Original signal; (b) Reconstruction with BP, $||\boldsymbol{f}_{BP} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.1582$, $t_{BP} = 1.56$ secs; (c) Reconstruction with BCS, $||\boldsymbol{f}_{BCS} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.0146$, $t_{BCS} = 0.63$ secs.

Figures 2(b-c) demonstrate the reconstruction results with BP and BCS, respectively. Because it is a noisy reconstruction problem, BP cannot recover the underlying sparse signal exactly. Consequently, the

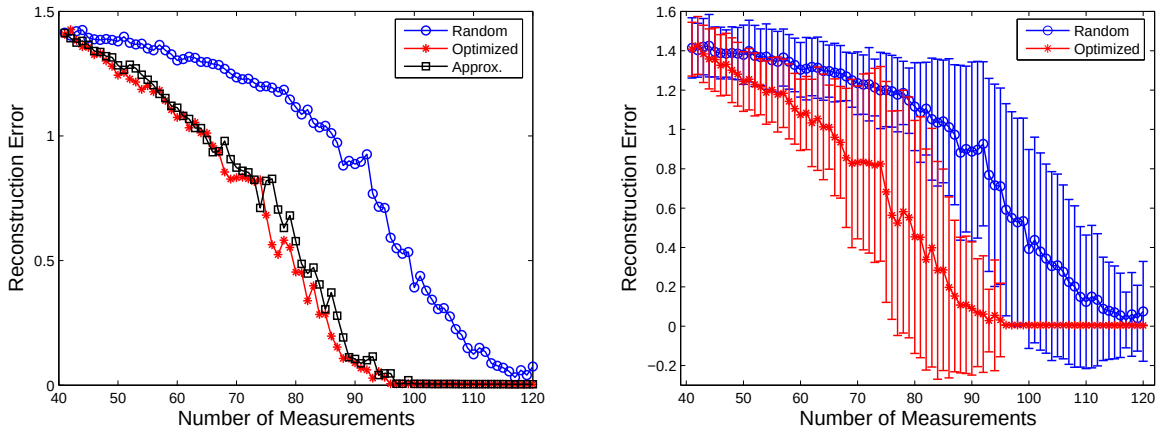

Fig. 3. Comparison of adaptive and random projections, with the first 40 projections performed randomly. (a) Reconstruction error of BCS with random projections, optimized projections (Sec. IV-A) and approximated projections (Sec. IV-B); the results are averaged over 100 runs; (b) the variances of the reconstruction error of BCS with random projections and optimized projections (Sec. IV-A); the variance for the approximate scheme in Sec. IV-B is very similar to that for Sec. IV-A, and thus is omitted to improve visibility.

BCS reconstruction is much cleaner than BP, as $M = 20$ spikes are correctly recovered with (about 10 times) smaller reconstruction error relative to BP. In addition, BCS yields "error-bars" for the estimated signal, indicating the confidence for the current estimation. Regarding the computation time, BCS also outperforms BP.

As discussed in Sec. IV, the Bayesian analysis also allows designing projection matrix $\mathbf{\Phi}$ for adaptive CS. In the second experiment, we use the same dataset as in Fig. 2 and study the performance of BCS for projection design. The initial 40 measurements are conducted by using the random projections as in Fig. 2, except that the rows of $\mathbf{\Phi}$ are normalized to 1.01 for the reasons discussed in Sec. IV-A. The remaining 80 measurements are sequentially conducted by optimized projections, with this compared to using random projections. In the experiment, after each projection vector $\boldsymbol{r}_{K+1}$ is determined, the associated reconstruction error is also computed. For the optimized projection, $\boldsymbol{r}_{K+1}$ is constructed by using the eigenvector of $\mathbf{\Sigma}$ that has the largest eigenvalue. When examining the approximate scheme discussed in Sec. IV-B, we used 10% of the average value of the diagonal elements of $(\alpha_0 \mathbf{\Phi}^T \mathbf{\Phi})$ for diagonal loading. Because of the randomness in the experiment (i.e., the initial 40 random projections and the empty-entries imputation for $\boldsymbol{r}_{K+1}$, etc.), we execute the experiment 100 times with the average performance reported in Fig. 3.

It is demonstrated in Fig. 3 that the reconstruction error of the optimized projection is much smaller

than that of the random projection, indicating the superior performance of this optimization. Further, the approximate scheme in Sec. IV-B yields results very comparable to the more-rigorous analysis in Sec. IV-A. This suggests that existing CS software may be readily modified to implement the optimization procedure, and yield results comparable to that of the full BCS solution.

We also note the following practical issue for implementation of an adaptive CS algorithm. Assume that an initial set of CS measurements are performed with a fixed set of projections, for which data $\boldsymbol{g}$ are measured. Based upon $\boldsymbol{g}$ and knowledge of the initial projections, there is a deterministic mapping to the next (optimized projection), with which the next CS measurement is performed. Consequently, although the adaptive projections are performed on the sensor, when performing signal reconstruction subsequently, the optimized projections that are performed at the sensor may be inferred offline, and therefore there is no need to send this information to the decoder. Consequently, the performance of optimized projections introduces no new overhead for storage of the compressive measurements $\boldsymbol{g}$ (we do not have to store the adaptively determined projections).

B. BCS vs. BP and StOMP

In the following set of experiments, the performance of BCS is compared to BP and StOMP on three example problems included in the *Sparselab* package that is available online at `http://sparselab.stanford.edu/`. Following the experiment setting in the package, all the projection matrix $\boldsymbol{\Phi}$ here are drawn from a uniform spherical distribution [7].

1) Bumps: Figure 4(a) shows the *Bumps* data used in [10], rendered with $N = 4096$ samples. Such signals are known to have wavelet expansions with relatively few significant coefficients. We applied a hybrid CS scheme [7] to signal reconstruction, with a coarsest scale $j_0 = 5$, and a finest scale $j_1 = 10$ on the "symmlet8" wavelet. For the CS measurements $K = 640$, with these measurements reflecting random linear combinations of the wavelet coefficients of Bumps. We compared the performance of BCS to that of BP and StOMP equipped with CFDR and CFAR thresholding [10]. Evidently, the accuracy of reconstruction is comparable for all the algorithms used, but the speed of BCS is between that of BP and the two StOMP implementations, with StOMP being the most efficient. However, as we have noted, the StOMP algorithm has several thresholding parameters to tune, whose values are critical to the performance of StOMP, while BCS has no such issue.

2) Random-Bars: Figure 5 shows the reconstruction results for *Random-Bars* that has been used in [7]. We used the Haar wavelet expansion, which is naturally suited to images of this type, with a coarsest scale $j_0 = 3$, and a finest scale $j_1 = 6$. Figure 5(a) shows the result of linear reconstruction with $K = 4096$

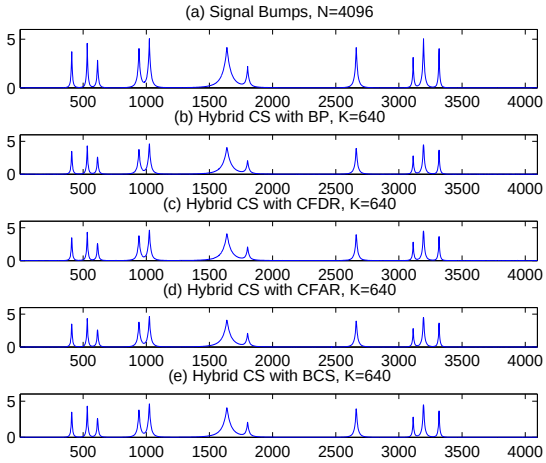

Fig. 4. Reconstruction of *Bumps* with hybrid CS. (a) Signal Bumps, with $N = 4096$ samples; (b) Reconstruction with BP, $||\boldsymbol{f}_{BP} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.0372$, $t_{BP} = 5.25$ secs; (c) Reconstruction with CFDR, $||\boldsymbol{f}_{CFDR} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.0375$, $t_{CFDR} = 0.83$ secs; (d) Reconstruction with CFAR, $||\boldsymbol{f}_{CFAR} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.0370$, $t_{CFAR} = 0.79$ secs; (e) Reconstruction with BCS, $||\boldsymbol{f}_{BCS} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.0365$, $t_{BCS} = 4.53$ secs.

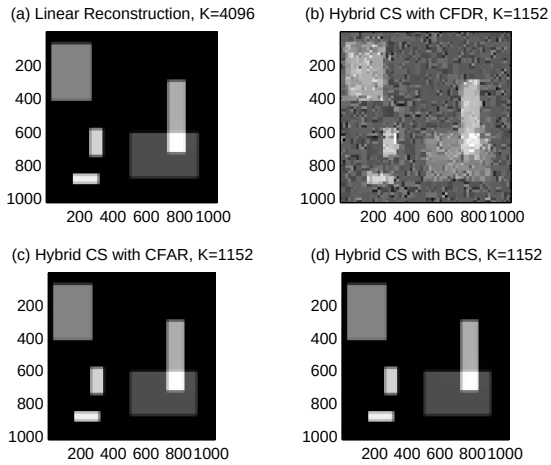

Fig. 5. Reconstruction of *Random-Bars* with hybrid CS. (a) Linear reconstruction from $K = 4096$ samples, $||\boldsymbol{f}_{LIN} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.2271$; (b) Reconstruction with CFDR, $||\boldsymbol{f}_{CFDR} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.5619$, $t_{CFDR} = 3.69$ secs; (c) Reconstruction with CFAR, $||\boldsymbol{f}_{CFAR} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.2271$, $t_{CFAR} = 4.67$ secs; (c) Reconstruction with BCS, $||\boldsymbol{f}_{BCS} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.2271$, $t_{BCS} = 15.41$ secs. BP took 108 secs with the reconstruction error 0.2279, which is not shown here.

samples, which represents the best performance that could be achieved by all the CS implementations used, whereas Figs. 5(b-d) have results for the hybrid CS scheme [7] with $K = 1152$ hybrid compressed samples. It is demonstrated that BCS and StOMP with CFAR yield the near optimal reconstruction error

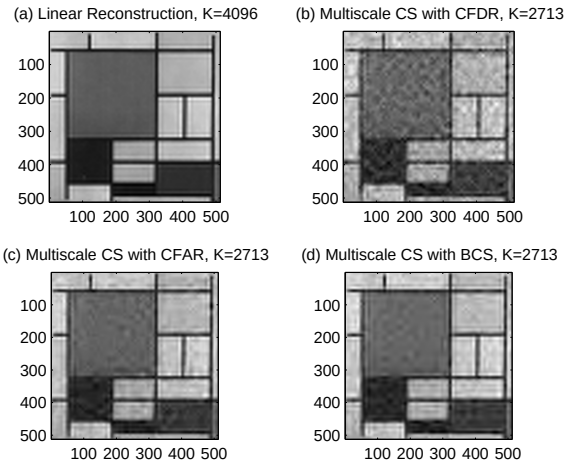

Fig. 6. Reconstruction of *Mondrian* with multiscale CS. (a) Linear reconstruction from $K = 4096$ samples, $||\boldsymbol{f}_{LIN} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.1333$; (b) Reconstruction with CFDR, $||\boldsymbol{f}_{CFDR} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.1826$, $t_{CFDR} = 10$ secs; (c) Reconstruction with CFAR, $||\boldsymbol{f}_{CFAR} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.1508$, $t_{CFAR} = 28$ secs; (c) Reconstruction with BCS, $||\boldsymbol{f}_{BCS} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2 = 0.1503$, $t_{BCS} = 18$ secs. BP took 162 secs with the reconstruction error 0.1416, which is not shown here.

(0.2271); among all the CS algorithms considered StOMP is the fastest one. However, as we have noted, the performance of StOMP strongly relies on the thresholding parameters selected. For the Random-Bars problem considered, the performance of StOMP with CFDR is very sensitive to its parameter-setting, with one typical example results shown in Fig. 5(b).

3) Mondrian: Figure 6 displays a photograph of a painting by Piet Mondrian, the Dutch neo-plasticist. Despite being a simple geometric example, this image still presents a challenge, as its wavelet expansion is not as sparse as the examples considered above. We used a multiscale CS scheme [7] for image reconstruction, with a coarsest scale $j_0 = 4$, and a finest scale $j_1 = 6$ on the "symmlet8" wavelet. Figure 6(a) shows the result of linear reconstruction with $K = 4096$ samples, which represents the best performance that could be achieved by all the CS implementations used, whereas Figs. 6(b-d) have results for the multiscale CS scheme with $K = 2713$ multiscale compressed samples. In the example results in Figs. 6(b-c), we used the same parameters-setting for StOMP as those used in the *Sparselab* package. It is demonstrated that all the CS implementations yielded a faithful reconstruction to the original image, while BCS produced the second smallest reconstruction error (0.1503) using the second smallest computation time (18 secs).

To understand why BCS is more efficient than StOMP on this problem, we checked the number of nonzero weights recovered by BCS and StOMP, with the results reported in Table I. Evidently, BCS

found the sparsest solution (with 615 nonzeros) relative to the two StOMP implementations, but yielded the smallest reconstruction error (0.1503). This indicates that although each iteration of StOMP allows multiple nonzero weights to be added into the "active set" [10], this process may be a too generous usage of weights without reducing the reconstruction error. The sparser solution of BCS is the likely explanation of its relative higher speed compared to StOMP in this example.

TABLE I

SUMMARY OF THE PERFORMANCES OF STOMP AND BCS ON MONDRIAN.

	StOMP with CFDR	StOMP with CFAR	BCS
# Nonzeros	1766	926	615
Reconst. Error	0.1826	0.1508	0.1503

VI. CONCLUSIONS

Compressive sensing has been considered from a Bayesian perspective, and it has been demonstrated that the conventional ℓ_1-regularized inversion for the underlying signal corresponds to a maximum *a posteriori* (MAP) solution, with a Laplace sparseness prior. Borrowing ideas from the machine-learning community, specifically the relevance vector machine (RVM) [12], the Laplace prior was replaced by a hierarchical prior, which yields a graphical model in which consecutive components within the graph are in the conjugate-exponential family. This framework is particularly attractive, for it allows efficient computation via such techniques as variational Bayesian analysis [15]. For the work of interest here we are most concerned about the uncertainty associated with the underlying signal weights, and therefore a type-II maximum-likelihood solution is performed [12]. This framework leads to a computationally efficient implementation, in which basis functions are added and deleted to the solution adaptively [18], [19]. We have found in practice that the results from this analysis are often sparser than existing related CS solutions [6], [10], and it is also important that there are no free parameters to be set. On the examples considered from the literature, we have found that the BCS solution typically has computation time comparable to state-of-the-art algorithms such as StOMP [10]; in some cases BCS is even faster than StOMP as a consequence of the improved sparsity of the BCS solution. However, on extremely large problems BCS should not be as fast as StOMP, since StOMP and the related algorithms only compute a point estimate of $\boldsymbol{w}$, while BCS finds the posterior density function of $\boldsymbol{w}$ (i.e., the covariance matrix $\boldsymbol{\Sigma}$ is also estimated).

As a consequence of the Bayesian formulation, we may ask questions that are not typically addressed in previous CS solutions, for which only a point estimate has been provided for the underlying weights. Specifically, since we have "error bars" on the inverted weights, we may ask what additional projections should be performed, with the objective of reducing inversion uncertainty. We have demonstrated that one may significantly accelerate the rate at which CS converges (use fewer CS measurements) if the projections are selected adaptively. As an important practical consideration, the optimal next projection may be inferred at the encoder and decoder based on the observed data, and therefore one need not transfer to the decoder the optimized projections. We have also implemented a simple approximation to the adaptive CS procedure, with which one may use the results from previous CS inversion algorithms, such as OMP [9] and StOMP [10]. It was demonstrated that this simple approximation yields adaptive CS results that are almost as good as those that come from the full Bayesian analysis. Therefore, the procedures presented here constitute a technique that may be employed within existing CS algorithms, to yield an immediate and marked improvement in performance.

Although we have demonstrated via experiments a significantly accelerated rate of convergence for adaptive CS relative to the conventional CS, a theoretical analysis of adaptive CS may be a worthy direction of future research. This can be an important complement to the existing analysis for the conventional CS [24], [25]. In addition, the Bayesian formalism allows convenient implementation of concepts that are not directly amenable to conventional CS techniques. For example, in the inversion procedure considered here the prior information that has been exploited is the fact that the signal is compressible in some basis. If additional prior information were available, for example based on previous relevant sensing experience, then this information may in principle be utilized within a refined prior. In future research we also plan to further exploit such flexibility of the Bayesian solution, with incorporation of additional "side information" [23] within the prior.

REFERENCES

[1] S. Mallat, *A wavelet tour of signal processing*, 2nd ed. Academic Press, 1998.

[2] I. Daubechies, *Ten lectures on wavelets*. SIAM, 1992.

[3] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Systems for Video Technology*, vol. 6, pp. 243–250, 1996.

[4] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Trans. Circuits Systems Video Technology*, vol. 14, pp. 1219–1235, Nov. 2004.

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

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

[7] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal Processing*, vol. 86, no. 3, pp. 549–571, Mar. 2006.

[8] S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.

[9] J. A. Tropp and A. C. Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," Apr. 2005, Preprint.

[10] 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.

[11] A. Papoulis and S. U. Pillai, *Probability, random variables and stochastic processes*, 4th ed. McGraw-Hill, 2002.

[12] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.

[13] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Processing*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[14] J. M. Bernardo and A. F. M. Smith, *Bayesian theory*. Wiley, 1994.

[15] C. M. Bishop and M. E. Tipping, "Variational relevance vector machines," in *Proc. of the 16th Conference on Uncertainty in Artificial Intelligence*, 2000, pp. 46–53.

[16] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1996.

[17] T. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, pp. 25–37, 2000.

[18] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," in *Advances in Neural Information Processing Systems (NIPS 14)*, 2002.

[19] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. of the Ninth International Workshop on Artificial Intelligence and Statistics*, C. M. Bishop and B. J. Frey, Eds., 2003.

[20] V. V. Fedorov, *Theory of optimal experiments*. Academic Press, 1972.

[21] D. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[22] S. Ji, B. Krishnapuram, and L. Carin, "Variational Bayes for continuous hidden Markov models and its application to active learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, pp. 522–532, Apr. 2006.

[23] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY: Wiley, 1991.

[24] E. Candès and T. Tao, "The Dantzig selector: statistical estimation when p is much larger than n," 2005, Preprint.

[25] J. Haupt and R. Nowak, "Signal reconstruction from noisy random projections," *IEEE Trans. Information Theory*, vol. 52, no. 9, pp. 4036–4048, Sept. 2006.