Bayesian Compressive Sensing

Shihao Ji, Ya Xue, and Lawrence Carin

Department of Electrical and Computer Engineering

Duke University, Durham, NC 27708-0291 USA

{shji,yx10,lcarin}@ece.duke.edu

EDICS: DSP-RECO, MAL-BAYL

Abstract

The data of interest are assumed to be represented as N-dimensional real vectors, and these vectors are compressible in some linear basis $\mathbf{B}$, implying that the signal can be reconstructed accurately using only a small number $M \ll N$ of basis-function coefficients associated with $\mathbf{B}$. 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 $\mathbf{B}$. 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), Experimental design, Adaptive compressive sensing, Bayesian model selection.

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, assuming that the data index represents time (it may similarly represent space). The localized properties of these orthonormal time-frequency atoms yields highly compact representations of many natural signals [1], [2]. Let the $N \times N$ matrix $\mathbf{B}$ represent a wavelet basis, with basis functions defined by associated columns; a general signal $\boldsymbol{f} \in \mathbb{R}^N$ may be represented as $\boldsymbol{f} = \mathbf{B}\boldsymbol{w}$, 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 $\hat{\boldsymbol{f}} = \mathbf{B}\hat{\boldsymbol{w}}$, 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 coding 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 [5], [6], introducing the field of compressive sensing (CS).

In its earliest form the relationship between the underlying signal $\boldsymbol{f}$ and the CS measurements $\boldsymbol{g}$ has been constituted through random projections [6], [7]. Specifically, assume that the signal $\boldsymbol{f}$ is compressible in some basis $\mathbf{B}$ (not necessarily a wavelet basis), the k-th CS measurement g_k (k-th 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 $\mathbf{B}$, i.e., $g_k = \boldsymbol{f}^T(\mathbf{B}\boldsymbol{r}_k)$, 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) [6], [7].

Based on the above discussion, the CS measurements may be represented as $\boldsymbol{g} = \mathbf{\Phi}\mathbf{B}^T\boldsymbol{f} = \mathbf{\Phi}\boldsymbol{w}$, where $\mathbf{\Phi} = [\boldsymbol{r}_1 \ldots \boldsymbol{r}_K]^T$ is a $K \times N$ matrix, assuming K random CS measurements are made. Since typically $K < N$ this amounts to having fewer measurements than degrees of freedom for the signal $\boldsymbol{f}$. Therefore, inversion for the N-weights represented by $\boldsymbol{w}$ (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 $\mathbf{B}$, then one may approximate

w accurately [5], [6]. A typical means of solving such an ill-posed problem, for which it is known that w is sparse, is via an ℓ_1-regularized formulation [6]

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

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 a point estimate of the weights 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 w should be sparse in the basis $\mathbf{B}$, data g are observed from compressive measurements, and the objective is to provide a posterior belief (density function) for the values of the weights w. Besides the improved accuracy over the point estimate (to be discussed in Sec. III-B), the Bayesian formalism, more importantly, provides a new framework that allows us to address a variety of issues that previously have not been addressed. Specifically, rather than providing a point (single) estimate for the weights w, a full posterior density function is provided, which yields "error bars" on the estimated f; these error bars may be used to give a sense of confidence in the approximation to 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 f; in addition, the Bayesian framework provides an estimate for the posterior density function of additive noise encountered when implementing the compressive measurements.

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 f. In Sec. IV we examine how this framework allows adaptive CS, whereby the aforementioned projections r_k are selected to optimize a (myopic) information measure. Example results on canonical data are presented in Sec. V, with comparisons to other 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 f is compressible in the basis $\mathbf{B}$. Therefore, let w_s represent an N-dimensional vector that is identical to the vector w for the M elements in 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

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

with $\boldsymbol{n}_e=\boldsymbol{\Phi}\boldsymbol{w}_e$. 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[1] with unknown variance σ^2. We therefore have the Gaussian likelihood model

$$p(\boldsymbol{g}|\boldsymbol{w}_s,\sigma^2)=(2\pi\sigma^2)^{-K/2}\exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{g}-\boldsymbol{\Phi}\boldsymbol{w}_s\|^2\right). \tag{4}$$

This 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. 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 [12], [13]:

$$p(\boldsymbol{w}|\lambda)=(\lambda/2)^N\exp(-\lambda\sum_{i=1}^{N}|w_i|), \tag{5}$$

where in (5) and henceforth we drop the subscript s on $\boldsymbol{w}$, recognizing that we are always interested in a sparse solution for the weights. Given the CS measurements $\boldsymbol{g}$, and assuming the likelihood function in (4), it is straightforward to demonstrate that the solution in (1) corresponds to a maximum *a posteriori* (MAP) estimate for $\boldsymbol{w}$ using the prior in (5) [13], [14].

---

[1]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, motivated for the reasons discussed above, and desirable from the standpoint of analysis.

III. ESTIMATE OF SPARSE WEIGHTS VIA RELEVANCE VECTOR MACHINE

A. Hierarchical Sparseness Prior

The above discussion connected conventional CS inversion for the weights $\boldsymbol{w}$ 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 σ^2. This is not readily accomplished using the Laplace prior directly, since the Laplace prior is not conjugate[2] to the Gaussian likelihood and hence the associated Bayesian inference may not be performed in closed form [12], [15].

This issue has been addressed previously in sparse Bayesian learning, particularly, with the relevance vector machine (RVM) [16]. Rather than imposing a Laplace prior on $\boldsymbol{w}$, in the RVM a hierarchical prior has been invoked, which has similar properties as the Laplace prior but allows convenient conjugate-exponential analysis. To see this, one first defines a zero-mean Gaussian prior on each element of $\boldsymbol{w}$:

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{N}(w_i|0, \alpha_i^{-1}), \tag{6}$$

with α_i the precision (inverse-variance) of a Gaussian density function. Further, a Gamma prior is considered over $\boldsymbol{\alpha}$:

$$p(\boldsymbol{\alpha}|a, b) = \prod_{i=1}^{N} \Gamma(\alpha_i|a, b). \tag{7}$$

By marginalizing over the hyperparameters $\boldsymbol{\alpha}$, the overall prior on $\boldsymbol{w}$ is then evaluated as

$$p(\boldsymbol{w}|a, b) = \prod_{i=1}^{N} \int_0^\infty \mathcal{N}(w_i|0, \alpha_i^{-1})\Gamma(\alpha_i|a, b)d\alpha_i. \tag{8}$$

The 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 [16]. With appropriate choice of a and b, the Student-t distribution is strongly peaked about $w_i = 0$, and therefore the prior in (8) favors most w_i being zero (i.e., it is a sparseness prior). Similarly, a Gamma prior $\Gamma(\alpha_0|c, d)$ is introduced on the inverse of the noise variance $\alpha_0 = 1/\sigma^2$.

To see the advantage of this hierarchical prior, 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

---

[2]In Bayesian probability theory, a class of prior probability distributions $p(\theta)$ is said to be conjugate to a class of likelihood functions $p(x|\theta)$ if the resulting posterior distributions $p(\theta|x)$ are in the same family as $p(\theta)$.
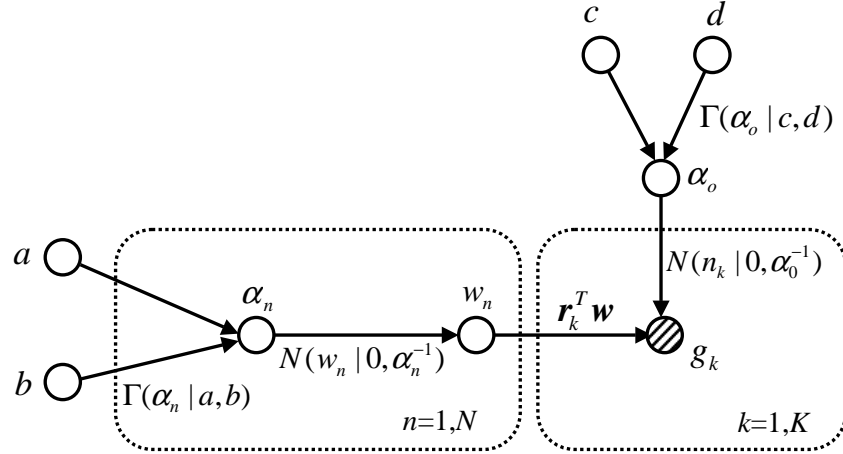
Fig. 1. Graphical model of the Bayesian CS formulation.

next parameter in the sequence. For all steps in Fig. 1, the density function for p_k is the conjugate 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). This structural form is very convenient for implementing iterative algorithms for evaluation of the posterior density function for w and α_0. For example, one may conveniently implement a Markov Chain Monte Carlo (MCMC) [17] or, more efficiently and approximately, a variational Bayesian (VB) analysis [18]. While the VB analysis is efficient relative to MCMC, in the RVM a type-II maximum-likelihood (ML) procedure is considered, with the objective of achieving highly efficient computations while still preserving accurate results.

As one may note, the Bayesian linear model considered in RVM is essentially one of the simplified models for Bayesian model selection [19]–[21]. Although more accurate models may be desired, the main motivation of adopting the RVM is due to its highly efficient computation as discussed below.

B. Bayesian CS Inversion via RVM

Assuming the hyperparameters α and α_0 are known, given the CS measurements g and the projection matrix Φ, the posterior for w can be expressed analytically as a multivariate Gaussian distribution with mean and covariance:

$$\mu = \alpha_0 \Sigma \Phi^T g, \tag{9}$$

$$\Sigma = (\alpha_0 \Phi^T \Phi + \mathbf{A})^{-1}, \tag{10}$$

where $\mathbf{A} = \text{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$. The associated "learning" problem, in the context of the RVM, thus becomes the search for the hyperparameters $\boldsymbol{\alpha}$ and α_0. In the RVM these hyperparameters are estimated from the data by performing a type-II ML (or evidence maximization) procedure [16]. Specifically, by marginalizing over the weights $\boldsymbol{w}$, the marginal likelihood for $\boldsymbol{\alpha}$ and α_0, or equivalently, its logarithm $\mathcal{L}(\boldsymbol{\alpha}, \alpha_0)$ can be expressed analytically as

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\alpha}, \alpha_0) &= \log p(\boldsymbol{g}|\boldsymbol{\alpha}, \alpha_0) = \log \int p(\boldsymbol{g}|\boldsymbol{w}, \alpha_0) p(\boldsymbol{w}|\boldsymbol{\alpha}) d\boldsymbol{w} \\
&= -\frac{1}{2}\left[K \log 2\pi + \log |\mathbf{C}| + \boldsymbol{g}^T \mathbf{C}^{-1} \boldsymbol{g} \right],
\end{aligned}
\tag{11}
$$

with $\mathbf{C} = \sigma^2 \mathbf{I} + \boldsymbol{\Phi} \mathbf{A}^{-1} \boldsymbol{\Phi}^T$. A type-II ML approximation employs the point estimates for $\boldsymbol{\alpha}$ and α_0 to maximize (11), which can be implemented via the EM algorithm (or other techniques) [16], to yield:

$$
\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2}, \qquad i \in \{1, 2, \ldots, N\},
\tag{12}
$$

where μ_i is the i-th posterior mean weight from (9) and we have defined the quantities $\gamma_i \triangleq 1 - \alpha_i \boldsymbol{\Sigma}_{ii}$, with $\boldsymbol{\Sigma}_{ii}$ the i-th diagonal element of the posterior weight covariance from (10). For the noise variance $\sigma^2 = 1/\alpha_0$, differentiation leads to the re-estimate:

$$
1/\alpha_0^{new} = \frac{\|\boldsymbol{g} - \boldsymbol{\Phi}\boldsymbol{\mu}\|_2^2}{K - \sum_i \gamma_i}.
\tag{13}
$$

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, which iterates between (9)-(10) and (12)-(13), until a convergence criterion has been satisfied. In this process, it is observed that many of the α_i tend to infinity (or are numerically indistinguishable from infinity given the machine precision) 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 for representation of $\boldsymbol{g}$, and the level of sparseness (size of M) is determined automatically (see [22] for an interesting explanation from a variational approximation perspective). It is also important to note that, as a result of the type-II ML estimate (11), the point estimates (rather than the posterior densities) of $\boldsymbol{\alpha}$ and α_0 are sought. Therefore, there is no need to set a, b, c and d on the Gamma hyperpriors. This is equivalent to setting a, b, c and d to zero, and thus uniform hyperpriors (over a *logarithmic* scale) on $\boldsymbol{\alpha}$ and α_0 have been invoked [16].

While it is useful to have a measure of uncertainty in the weights $\boldsymbol{w}$, the quantity of most interest is the signal $\boldsymbol{f} = \mathbf{B}\boldsymbol{w}$. Since $\boldsymbol{w}$ is drawn from a multivariate Gaussian distribution with mean and covariance defined in (9)-(10), the posterior density function on $\boldsymbol{f}$ is also a multivariate Gaussian distribution with

mean and covariance:

$$\mathrm{E}(\boldsymbol{f}) \;=\; \mathbf{B}\boldsymbol{\mu}, \tag{14}$$

$$\mathrm{Cov}(\boldsymbol{f}) \;=\; \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^{T}. \tag{15}$$

The diagonal elements of the covariance matrix in (15) provide "error bars"[3] on the accuracy of the inversion of $\boldsymbol{f}$, as represented in terms of its mean in (14).

While the iterative algorithm described above has been demonstrated to yield a highly accurate sparse linear-regression representation [16], we note the following practical limitation. When evaluating (10) one must invert matrices of size $N \times N$: an $\mathcal{O}(N^3)$ operation[4], thereby making this approach relatively slow for data $\boldsymbol{f}$ of large dimension N (at least for the first few iterations). This motivates development of a fast RVM algorithm with the objective of achieving highly efficient computations that are comparable to existing CS algorithms (e.g., OMP [9] and StOMP [10]).

Fortunately, this fast RVM algorithm has been developed in [26], [27] by analyzing the properties of the marginal likelihood function in (11). This enables a principled and efficient sequential addition and deletion of candidate basis function (columns of $\boldsymbol{\Phi}$) to monotonically maximize the marginal likelihood. We omit the detailed discussion of this fast algorithm, and refer the reader to [26], [27] for more details. We here only briefly summarize some of its key properties. Compared to the iterative algorithm presented above, the fast algorithm operates in a constructive manner, i.e., sequentially adds (or deletes) candidate basis function to the model until all M "relevant" basis functions (for which the associated weights are nonzero) have been included. Thus, the complexity of the algorithm is more related to M than N. Further, by exploiting the matrix inverse identity, the inverse operation in (10) has been implemented by an iterative update formula with reduced complexity. Detailed analysis of this algorithm shows that it has complexity $\mathcal{O}(NM^2)$, which is more efficient than the original RVM, especially when the underlying signal is truly sparse ($M \ll N$).

In contrast to other CS algorithms (e.g., OMP [9] and StOMP [10], in which basis functions once added are never removed), the fast RVM algorithm has the operation of deleting a basis function from the model (i.e., setting the corresponding $\alpha_i = \infty$). This deletion operation allows the fast algorithm to

---

[3]While previous works [23], [24] in CS do obtain ℓ_2 error bounds for function estimates, the "error bars" may be more useful from a practical standpoint as discussed in the next section.

[4]A simple modification to (10) is available from [25] by exploiting the matrix inverse identity, which leads to an $\mathcal{O}(K^3)$ operation per iteration. Nonetheless, the iterative (EM) implementation still does not scale well.

maintain a more concise signal representation and is likely one of the explanations for the improvement in sparsity demonstrated in the experiments (see Sec. V).

In addition, recent theoretical analysis of the RVM [28], [29] indicates that the RVM provides a tighter approximation to the ℓ_0-norm sparsity measure than the ℓ_1-norm, and prove that even in the worst-case scenario, the RVM still outperforms the most widely used sparse representation algorithms, including BP [8] and OMP [9]. Although these studies are based on the iterative (EM) implementation of the RVM, they indeed shed light on the fast implementation considered here, since both implementations are based on the same cost function (11). Our empirical study in Sec. V is also consistent with these theoretical results. Nonetheless, rigorous analysis of this fast algorithm remains worthy of further inquiry.

IV. ADAPTIVE COMPRESSIVE SENSING

A. Selecting Projections to Reduce Signal Uncertainty

In the original CS construction [6], [7], the projections represented by the matrix $\mathbf{\Phi}$ were constituted via i.i.d. realizations of an underlying random variable. In addition, previous CS algorithms [8]–[10] focused on estimating $\boldsymbol{w}$ (and hence $\boldsymbol{f}$) 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 $\mathbf{\Phi}$ was previously not feasible. The Bayesian CS (BCS) algorithm (in this case the fast RVM algorithm) discussed in Sec. III-B, allows efficient computation of $\boldsymbol{f}$ and associated error bars, as defined by (14) and (15), and therefore one may consider the possibility of adaptively selecting projection $\boldsymbol{r}_{K+1}$, with the goal of reducing uncertainty. Such a framework has been previously studied in the machine learning community under the name of experimental design or active learning [30]–[32]. 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 $\mathrm{E}(\boldsymbol{f}) = \mathbf{B}\boldsymbol{\mu}$ and covariance $\mathrm{Cov}(\boldsymbol{f}) = \mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^T$. The differential entropy [33] for $\boldsymbol{f}$ therefore satisfies:

$$
\begin{aligned}
h(\boldsymbol{f}) &= -\int p(\boldsymbol{f}) \log p(\boldsymbol{f}) \, d\boldsymbol{f} = \frac{1}{2} \log |\mathbf{B}\boldsymbol{\Sigma}\mathbf{B}^T| + const = \frac{1}{2} \log |\boldsymbol{\Sigma}| + const \\
&= -\frac{1}{2} \log |\mathbf{A} + \alpha_0 \mathbf{\Phi}^T \mathbf{\Phi}| + const,
\end{aligned}
\tag{16}
$$

where $const$ is independent of the projection matrix $\mathbf{\Phi}$. Recall that $\mathbf{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_N)$, 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)[5].

We may now ask which new projection $\boldsymbol{r}_{K+1}$ would be optimal for minimizing the differential entropy in (16). 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 vector, via the matrix determinant identity, we have

$$h_{new}(\boldsymbol{f}) = h(\boldsymbol{f}) - \frac{1}{2}\log\left(1 + \alpha_0 \boldsymbol{r}_{K+1}^T \boldsymbol{\Sigma} \boldsymbol{r}_{K+1}\right), \tag{17}$$

where α_0 and $\boldsymbol{\Sigma}$ are based on estimates found using the previous K measurements. To minimize h_{new}, the next projection $\boldsymbol{r}_{K+1}$ should hence be designed to maximize $\boldsymbol{r}_{K+1}^T \boldsymbol{\Sigma} \boldsymbol{r}_{K+1}$. Since

$$\boldsymbol{r}_{K+1}^T \boldsymbol{\Sigma} \boldsymbol{r}_{K+1} = \boldsymbol{r}_{K+1}^T \mathrm{Cov}(\boldsymbol{w}) \boldsymbol{r}_{K+1} = \mathrm{Var}(g_{K+1}), \tag{18}$$

this is equivalent to maximizing the variance of the *expected* measurement g_{K+1}. 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.

The adaptive framework provides an attractive setting for selection of the next projection $\boldsymbol{r}_{K+1}$, with the goal of optimizing – in a one-look-ahead (myopic) sense – the rate at which the uncertainty in $\boldsymbol{f}$ diminishes [30]–[32]. There are multiple ways this may be utilized in practice. If it is possible to design new projection $\boldsymbol{r}_{K+1}$ 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 $\mathbf{L}$ of possible next projections, with $\boldsymbol{r}_{K+1}$ selected from $\mathbf{L}$ with the goal of maximizing (18). 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 $\mathbf{L}$ may be more useful in practice, and this remains an important direction for future research.

We also note the following practical issue for implementation of adaptive CS. 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 optimized projections are performed on the sensor, when performing signal reconstruction subsequently,

---

[5]In practice, many of the α_i have the value of infinity (or exceed the machine precision), indicating the corresponding basis functions in $\boldsymbol{\Phi}$ are excluded for sparse representation. Therefore, when evaluating (16), both $\mathbf{A}$ and $\boldsymbol{\Phi}$ only employ elements corresponding to the basis functions selected by BCS, and they are thus reduced in general to small matrices.

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 g (i.e., we do not have to store the adaptively determined projections).

An additional issue needs to be clarified if the eigenvector of $\mathbf{\Sigma}$ is used for the next projection r_{K+1}. Due to the sparse Bayesian solution, $\mathbf{\Sigma}$ only employs elements corresponding to the associated nonzero components of w found based on BCS (i.e., $\mathbf{\Sigma}$ is reduced in general to a small matrix). Thus, when constructing the next projection based on the eigenvector, some entries of r_{K+1} will be empty. If we impute all the empty entries with zero, we are under the risk of being wrong. The initial estimate of w can be inaccurate; if we impute all the empty entries with zero, the estimate of w may be always biased and has no chance to be corrected, since the corresponding contributions from underlying true w are always ignored. To mitigate this problem, we impute the empty entries with random samples drawn i.i.d. from a Gaussian distribution $\mathcal{N}(0, 1)$. After the imputation, we re-scale the ℓ_2-norm of the imputed values to 0.14. By doing so, we utilize the optimized projection and meanwhile allow some contributions from the empty entries. Overall, the final projection r_{K+1} has the magnitude $\|r_{K+1}\|_2 \approx 1.01$.

B. Approximate Adaptive CS

The "error bars" on the estimated signal 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 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 established fast CS algorithms. In this section, we consider one possible approximate scheme for adaptive CS, and show that the adaptive CS and its approximate scheme may be only amenable to the Bayesian analysis.

The uncertainty in f and the adaptive algorithm in (18) rely on computation of the covariance matrix $\mathbf{\Sigma} = (\alpha_0 \mathbf{\Phi}^T \mathbf{\Phi} + \mathbf{A})^{-1}$. Since $\mathbf{\Phi}$ is assumed known (but which basis functions have been selected by BCS are unknown), this indicates that what is needed are estimates for α_0 and $\boldsymbol{\alpha}$, the latter required for the diagonal matrix $\mathbf{A}$. Concerning the diagonal matrix $\mathbf{A}$, it may be viewed from a signal processing standpoint as a regularization of the matrix $(\alpha_0 \mathbf{\Phi}^T \mathbf{\Phi})$, to assure that the matrix inversion is well-posed. While the Bayesian analysis in Sec. III indicates that the loading represented by $\mathbf{A}$ should be non-uniform, we may simply make $\mathbf{A}$ diagonalized uniformly, with value corresponding to a small fraction

of the average value of the diagonal elements of $(\alpha_0 \mathbf{\Phi}^T \mathbf{\Phi})$, i.e.,

$$
\begin{aligned}
\hat{\mathbf{\Sigma}} &= \left(\alpha_0 \mathbf{\Phi}^T \mathbf{\Phi} + \frac{\epsilon}{N_s} \text{trace}(\alpha_0 \mathbf{\Phi}^T \mathbf{\Phi}) \mathbf{I} \right)^{-1} \\
&= \alpha_0^{-1} \left(\mathbf{\Phi}^T \mathbf{\Phi} + \frac{\epsilon}{N_s} \text{trace}(\mathbf{\Phi}^T \mathbf{\Phi}) \mathbf{I} \right)^{-1},
\end{aligned}
\tag{19}
$$

where ϵ is a small positive value (e.g., $\epsilon = 0.1$), and N_s is the number of basis functions selected by BCS based on the current CS measurements. Since we are only interested in the eigenvectors of $\hat{\mathbf{\Sigma}}$, α_0 in (19) can be ignored for the computation of eigen-decomposition. Therefore, for an approximate adaptive CS, what is needed is only the basis functions selected by BCS, with which constitute the projection matrix $\mathbf{\Phi}$ in (19).

In the derivation of (19), we assume that the diagonal elements of $\mathbf{A}$ are relatively uniform, such that $\mathbf{A}$ can be approximated by a *uniform* diagonal matrix. While this assumption is typically valid for BCS, there is no guarantee for other CS algorithms, since other CS algorithms may select basis functions that are distinct from those selected by BCS. In Sec. V, when presenting example results, we make comparisons between the rigorous implementation of adaptive CS presented in Sec. IV-A and the approximate scheme discussed here, as applied to BCS and OMP[6]. As demonstrated, both the rigorous implementation and the approximate scheme succeed in BCS, while the approximate scheme fails in OMP. Intuitively, this is because the basis functions selected by OMP are different from those selected by BCS. Compared to BCS, some OMP-selected basis functions should be removed. Therefore, from the Bayesian analysis standpoint, the corresponding α_i should be infinity, and thus the matrix $\mathbf{A}$ cannot be approximated by a uniform diagonal matrix. These comparisons suggest that the adaptive CS developed in Sec. IV-A may be only amenable to the Bayesian analysis, while it may not be feasible for other CS algorithms, indicating the adaptive CS may be one of the unique advantages of BCS over other CS algorithms.

V. EXAMPLE RESULTS

We test the performances of BCS and adaptive CS on several example problems considered widely in the CS literature, with comparisons (when appropriate) made to BP [8], OMP [9] and StOMP [10]. While BP is a relatively computationally expensive algorithm that involves linear programming, OMP is a fast greedy strategy that iteratively selects basis functions most aligned with the current residual, and StOMP is an extension of OMP and may be one of the state-of-the-art fast CS algorithms. In the

---

[6]OMP outputs both the weights and the indices of the selected basis functions. With these selected basis functions (which form $\mathbf{\Phi}$), we can compute an approximate covariance matrix $\hat{\mathbf{\Sigma}}$ (19), from which we then compute the eigenvector.

experiments, all the computations were performed on a 3.4GHz Pentium machine. The Matlab code is available online at `http://www.ece.duke.edu/~shji/BCS.html`.

A. 1D Signals

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 magnitude. 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/`, and the BP parameters were set as those suggested by ℓ_1-magic.

Figures 2(b-c) demonstrate the reconstruction results with BP and BCS, respectively. Due to noisy measurements, BP cannot recover the underlying sparse signal exactly, nor can BCS. However, the 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 (19), we set $\epsilon = 0.1$ for diagonal loading. Because of the randomness in the experiment (i.e., the generation of the original spike signal, 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 and variance reported in Figs. 3(a-b), respectively.

It is demonstrated in Figs. 3(a-b) that the reconstruction error of the optimized projection is much
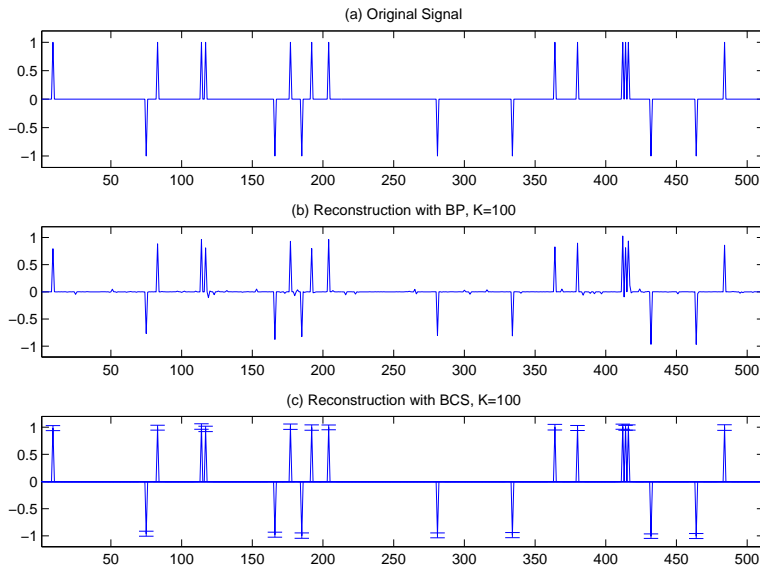
Fig. 2. Reconstruction of *uniform Spikes* for $N = 512$, $M = 20$, $K = 100$. (a) Original signal; (b) Reconstruction with BP, $\|\boldsymbol{f}_{BP} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.1582$, $t_{BP} = 1.66$ secs; (c) Reconstruction with BCS, $\|\boldsymbol{f}_{BCS} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.0146$, $t_{BCS} = 0.46$ secs.

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 rigorous implementation in Sec. IV-A, suggesting that this approximate scheme may be well-suited for BCS.

However, to make a meaningful conclusion, we still have two questions to address. First, the spike signal that we have considered above is exactly the case for which the nonzero entries of $\boldsymbol{w}$ have the same magnitude, and thus seems well-suited to the uniform loading assumption. Second, besides BCS, one may ask whether other CS algorithms may be modified to implement this approximate scheme, with the same success as BCS.

To answer the first question, we execute the same experiment as above but on a non-uniform spike signal as shown in Fig. 4(a). To make the comparison meaningful, the signal-to-noise-ratio (SNR) of the both types of spike signals are fixed the same. The results on the non-uniform spike signal are shown in Fig. 4 and Figs. 3(c-d), from which similar conclusions as for the uniform case can be made, indicating that the uniform loading assumption is generally applicable for BCS.

It is worthwhile to point out some notable observations from Fig. 3 regarding the performance of adaptive CS as compared to conventional CS. Specifically, Theorem 2 of [6] suggests that adaptive design of projections is of minimal help over (nonadaptive) random projections. Derived from information-based
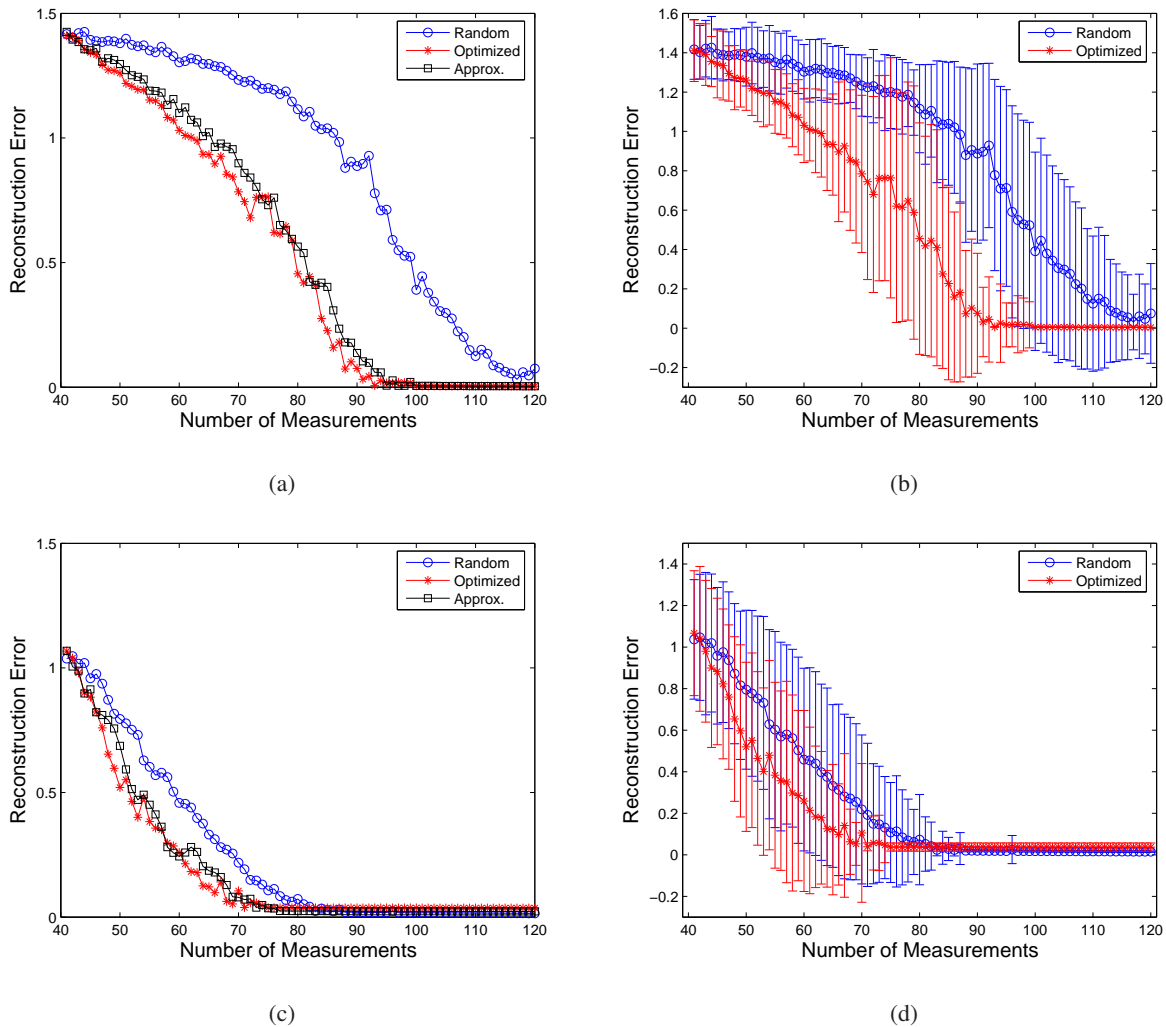
Fig. 3. Comparison of adaptive CS and conventional CS on uniform spikes and non-uniform spikes by using BCS; the results are averaged over 100 runs. (a, c) Reconstruction error of BCS with random projections, optimized projections (Sec. IV-A) and approximate projections (Sec. IV-B); (b, d) the variances of the reconstruction error of BCS with random projections and optimized projections (Sec. IV-A); the variance for the approximate projections (Sec. IV-B) is very similar to that of optimized projections, and thus is omitted to improve visibility. (a, b) are the results on uniform spikes (as in Fig. 2); (c, d) are the results on non-uniform spikes (as in Fig. 4). Note that the error bars (one standard deviations) in (c, d) only show how tight the errors are around their mean values, and do not indicate the errors can be negative.

complexity, Theorem 2 of [6] shows that

$$E^{\text{nonadapt}} \leq 2^{1/p} \cdot E^{\text{adapt}}, \tag{20}$$

where E^{method} denotes the minimax ℓ_2 reconstruction error of the method (adaptive or nonadaptive) adopted, and $p \in (0, 1]$ describes the compressibility of the underlying signal. In the most common case
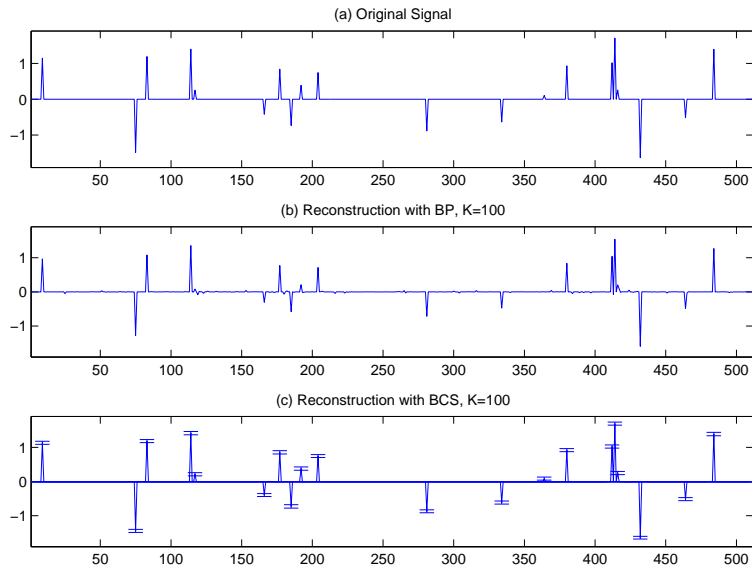
Fig. 4. Reconstruction of *non-uniform Spikes* for $N = 512$, $M = 20$, $K = 100$. (a) Original signal; (b) Reconstruction with BP, $\|\boldsymbol{f}_{BP} - \boldsymbol{f}\|_2/\|\boldsymbol{f}\|_2 = 0.1375$, $t_{BP} = 1.89$ secs; (c) Reconstruction with BCS, $\|\boldsymbol{f}_{BCS} - \boldsymbol{f}\|_2/\|\boldsymbol{f}\|_2 = 0.0178$, $t_{BCS} = 0.27$ secs.

$p = 1$, this inequality (20) elucidates that by using optimized projection at most 50% reduction in error can be attained as compared to random projection. Not surprisingly, our results in Fig. 3 is consistent with this conclusion. We note that 50% reduction in error may be not remarkable in theory (or from a mathematical standpoint), but from a practical engineering standpoint 50% error reduction is often significant.

As a secondary point, we also observe the following notable differences between the performances of BCS as applied to the uniform spikes and the non-uniform spikes. Comparing Fig. 3(a) to Fig. 3(c), for a given number of CS measurements, the reconstruction error on the non-uniform spikes is (much) smaller than that on the uniform spikes. Evidently, this observation is consistent with some of the theoretical analysis from [29], i.e., uniform weights offer the worst-case scenario for sparse signal reconstruction, and "the more diverse the weights magnitudes, the better the chances we have of learning the optimal solution".

To address the second question above, we test the approximate adaptive CS scheme as applied to OMP, with the results on the uniform spikes and the non-uniform spikes shown in Fig. 5. It is demonstrated in Fig. 5 that in both cases the approximate scheme with OMP are unsuccessful. Intuitively, this is because the basis functions selected by OMP are different from those selected by BCS. Compared to BCS, some OMP-selected basis functions should be removed. Therefore, from the Bayesian analysis standpoint, the

corresponding α_i should be infinity, and thus the matrix $\mathbf{A}$ cannot be approximated by a uniform diagonal matrix.

In summary, the empirical study presented above suggests that (i) the adaptive CS developed in Sec. IV-A outperforms conventional CS, and this improvement is more remarkable for signals with uniform weights; and (ii) the adaptive CS may be only amenable to the Bayesian analysis, while it may not be feasible for other CS algorithms (e.g., OMP), indicating a unique advantage of BCS over other CS algorithms. For these reasons, in the experiments that follow, when the adaptive CS is applied, we only consider the rigorous implementation presented in Sec. IV-A, with this a direct benefit from BCS.

| (a) | (b) |
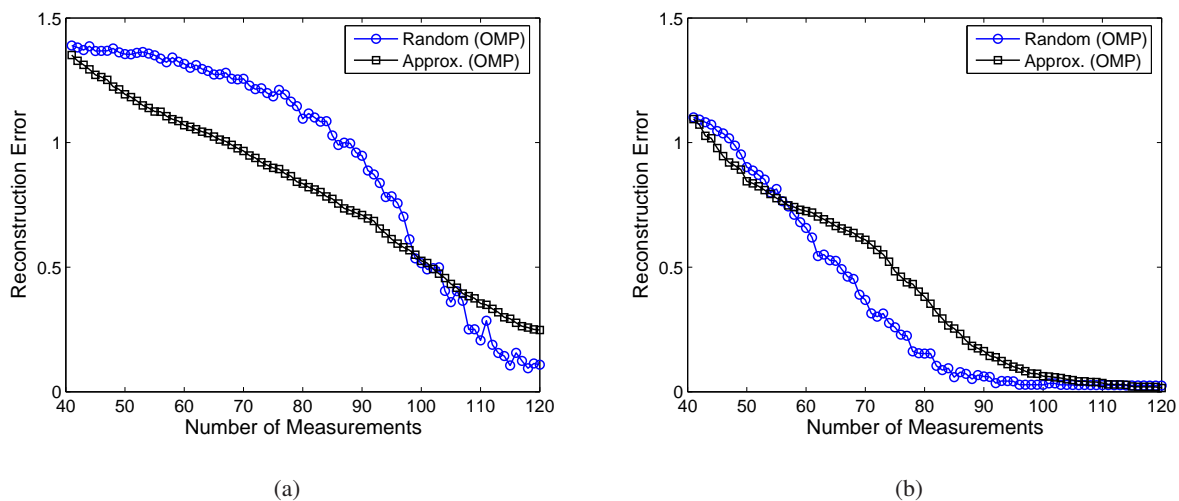
Fig. 5. Comparison of the approximate adaptive CS (Approx.) and conventional CS (Random) by using OMP on (a) uniform spikes (as in Fig. 2), and (b) non-uniform spikes (as in Fig. 4). The results are averaged over 100 runs.

B. 2D Images

In the following set of experiments, the performance of BCS is compared to BPDN (the noise-aware version of BP) and StOMP (equipped with CFDR and CFAR thresholding) on two 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]. For completeness, we also test the performances of adaptive CS on these two example images as compared to conventional CS.

1) Random-Bars: Figure 6 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 6(a) shows the result of linear reconstruction (i.e., the inverse wavelet transform) 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 hybrid CS scheme (i.e., the CS measurements are made only on the fine-scale coefficients; no compression on the coarsest-scale coefficients) [7] with $K = 1216$ hybrid compressed samples. It is demonstrated that BCS and StOMP with CFAR yield the near optimal reconstruction error (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 result shown in Fig. 6(b).
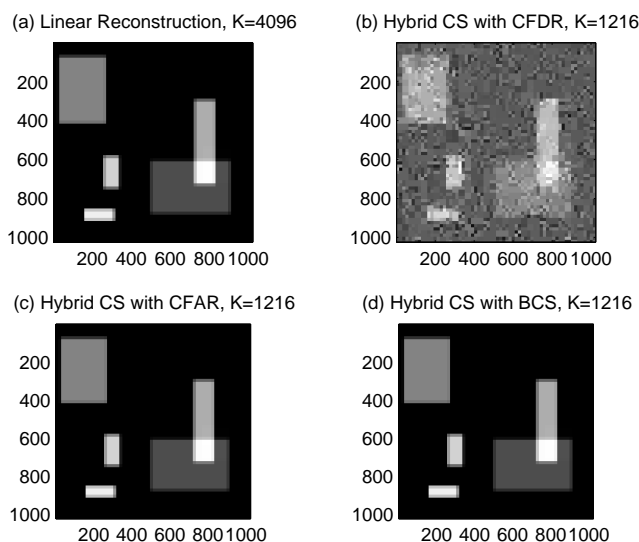
(a) Linear Reconstruction, K=4096

(b) Hybrid CS with CFDR, K=1216

(c) Hybrid CS with CFAR, K=1216

(d) Hybrid CS with BCS, K=1216

Fig. 6. 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, $\|\boldsymbol{f}_{CFDR} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.5619$, $t_{CFDR} = 3.15$ secs; (c) Reconstruction with CFAR, $\|\boldsymbol{f}_{CFAR} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.2271$, $t_{CFAR} = 4.38$ secs; (c) Reconstruction with BCS, $\|\boldsymbol{f}_{BCS} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.2271$, $t_{BCS} = 8.55$ secs. BP (ℓ_1) took 114 secs with the reconstruction error 0.2279, which is not shown here.

2) Mondrian: Figure 7 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 7(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. 7(b-d) have results for the multiscale CS scheme with $K = 2713$ multiscale compressed samples. In the example results in Figs. 7(b-c), we used the same parameter-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 (0.1498) using the second smallest computation time (15 secs).
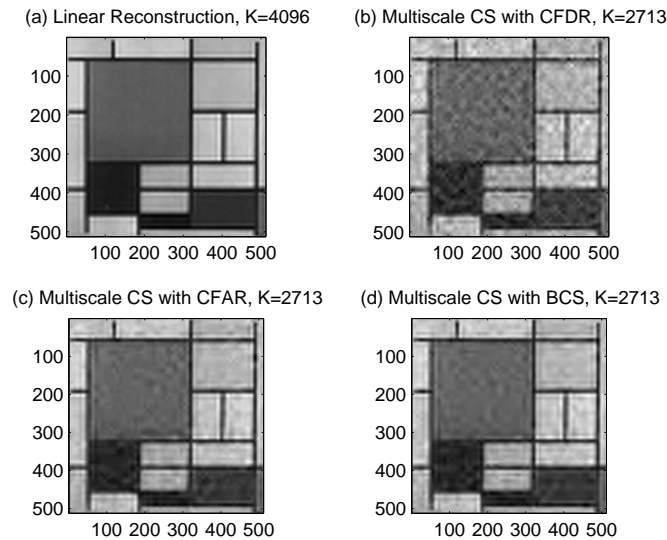
Fig. 7. Reconstruction of *Mondrian* with multiscale CS. (a) Linear reconstruction from $K = 4096$ samples, $\|\boldsymbol{f}_{lin} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.1333$; (b) Reconstruction with CFDR, $\|\boldsymbol{f}_{CFDR} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.1826$, $t_{CFDR} = 10$ secs; (c) Reconstruction with CFAR, $\|\boldsymbol{f}_{CFAR} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.1508$, $t_{CFAR} = 28$ secs; (c) Reconstruction with BCS, $\|\boldsymbol{f}_{BCS} - \boldsymbol{f}\|_2 / \|\boldsymbol{f}\|_2 = 0.1498$, $t_{BCS} = 15$ secs. BP (ℓ_1) took 162 secs with the reconstruction error 0.1416, which is not shown here.

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 751 nonzeros) relative to the two StOMP implementations, but yielded the second smallest reconstruction error (0.1498). 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.

Finally, the performances of adaptive CS as compared to conventional CS are provided in Figs. 8(a-b), for *Random-Bars* and *Mondrian*, respectively. The adaptive CS consistently outperforms conventional CS

TABLE I

SUMMARY OF THE PERFORMANCES OF BP, STOMP AND BCS ON MONDRIAN.

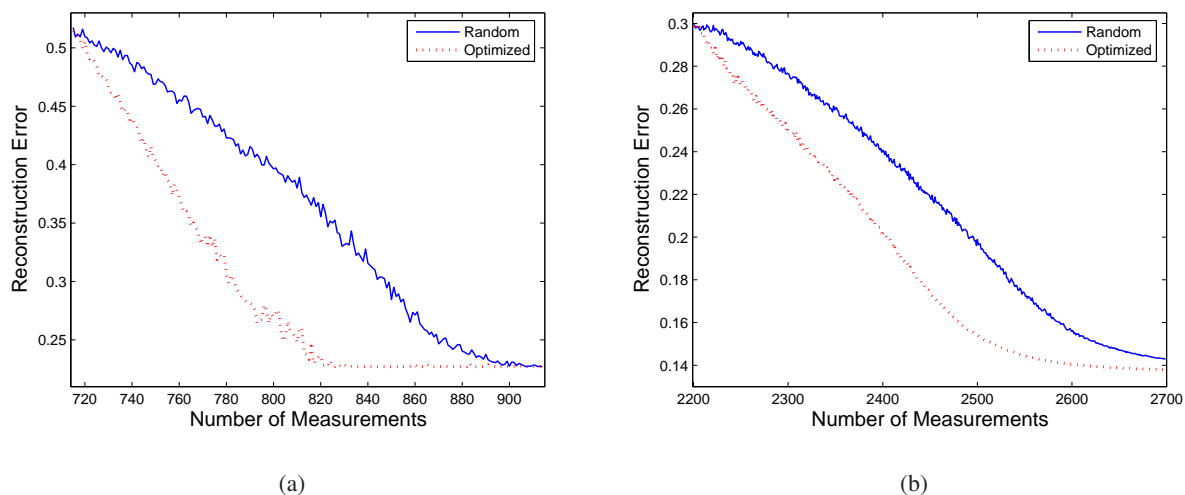|	BP	CFDR	CFAR	BCS
# Nonzeros	3840	1766	926	751
Time (secs)	162	10	28	15
Reconst. Error	0.1416	0.1826	0.1508	0.1498

in all the cases considered.

(a)   (b)

Fig. 8.   Comparison of adaptive CS (Optimized) and conventional CS (Random) on (a) Random-Bars, and (b) Mondrian. The results are averaged over 100 runs.

VI. CONCLUSIONS

Compressive sensing has been considered from a Bayesian perspective. It has been demonstrated that by utilizing the previously studied relevance vector machine (RVM) from the sparse Bayesian learning literature, problems in CS can be solved more effectively. In practice, we have found that the results from this Bayesian analysis are often sparser than existing CS solutions [8], [10]. On the examples considered from the literature, BCS typically has computation time comparable to the state-of-the-art algorithms such as StOMP [10]; in some cases, BCS is even faster as a consequence of the improved sparsity. We have also considered adaptive CS by optimizing the projection matrix Φ. Empirical studies demonstrate

a significantly accelerated rate of convergence compared to the original CS construction. Finally, we have also demonstrated that the adaptive CS may be only amenable to the Bayesian analysis, while it may not be feasible for other CS algorithms, indicating a unique advantage of BCS over other CS algorithms.

There is a clear connection between CS and regression shrinkage and selection via the Lasso [14], [34] as both focus on solving the same objective function (1). Research on Lasso has produced algorithms that might have some relevance to BCS. Besides this, other possible areas of future research may include (i) even faster sparse Bayesian learning algorithms, as dealing with images is a high-dimensional problem, (ii) simultaneous inversion of multiple data sets, borrowing ideas from multi-task learning [35], and (iii) a theoretical analysis of adaptive CS, as this could be an important complement to the existing analysis for the conventional CS formulation (e.g., [23], [24]).

ACKNOWLEDGMENTS

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

[13] M. Figueiredo, "Adaptive sparseness using Jeffreys prior," in *Advances in Neural Information Processing Systems (NIPS 14)*, 2002.

[14] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc B.*, vol. 58, no. 1, pp. 267–288, 1996.

[15] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, 2nd ed. CRC Press, 2003.

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

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

[18] 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.

[19] E. I. George and R. E. McCulloch, "Approaches for Bayesian variable selection," *Statistica Sinica*, vol. 7, pp. 339–373, 1997.

[20] E. I. George and D. P. Foster, "Calibration and empirical Bayes variable selection," *Biometrika*, vol. 87, no. 4, pp. 731–747, 2000.

[21] C. P. Robert, *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation*, 2nd ed. Springer, 2001.

[22] D. Wipf, J. Palmer, and B. Rao, "Perspectives on sparse Bayesian learning," in *Advances in Neural Information Processing Systems (NIPS 16)*, 2004.

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

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

[25] 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.

[26] A. C. Faul and M. E. Tipping, "Analysis of sparse Bayesian learning," in *Advances in Neural Information Processing Systems (NIPS 14)*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds., 2002, pp. 383–389.

[27] 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.

[28] D. P. Wipf and B. D. Rao, "ℓ_0-norm minimization for basis selection," in *Advances in Neural Information Processing Systems (NIPS 17)*, 2005.

[29] ——, "Comparing the effects of different weight distributions on finding sparse representations," in *Advances in Neural Information Processing Systems (NIPS 18)*, 2006.

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

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

[32] 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.

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

[34] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[35] R. Caruana, "Multitask learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.