Bayesian Compressive Sensing and Projection Optimization

Shihao Ji SHJI@ECE.DUKE.EDU
Lawrence Carin LCARIN@ECE.DUKE.EDU
Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708 USA

Abstract

This paper introduces a new problem for which machine-learning tools may make an impact. The problem considered is termed "compressive sensing", in which a real signal of dimension N is measured accurately based on $K \ll N$ real measurements. This is achieved under the assumption that the underlying signal has a sparse representation in some basis (e.g., wavelets). In this paper we demonstrate how techniques developed in machine learning, specifically sparse Bayesian regression and active learning, may be leveraged to this new problem. We also point out future research directions in compressive sensing of interest to the machine-learning community.

1. 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 [Mallat, 1998], 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 [Mallat, 1998]. 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 [Pearlman et al., 2004].

In conventional applications one first measures the N-dimensional signal $\boldsymbol{f}$, $\boldsymbol{f}$ is then transformed into a new basis where the signal is sparse, and the new basis coefficients $\boldsymbol{w}$ are then quantized [Pearlman et al., 2004]. 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 [Candès et al., 2006, Donoho, 2006], 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 [Candès et al., 2006, Donoho, 2006]. 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 a "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) [Tsaig & Donoho, 2006].

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 [Donoho, 2006]

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

where the scalar ρ controls the relative importance ap-

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

plied to the Euclidian error and the sparseness term. This basic framework has been the starting point for several recent CS inversion algorithms, including linear programming [Chen et al., 1999] and greedy algorithms [Tropp & Gilbert, 2005, Donoho et al., 2006], for a point estimate of the weights $\boldsymbol{w}$.

In the discussion that follows we demonstrate that the solution to this problem may exploit many of the tools developed recently in the machine-learning community, specifically sparse Bayesian regression [Tipping, 2001, Wipf et al., 2004] and active learning [Fedorov, 1972, MacKay, 1992]. Moreover, the results of this machine-learning-based CS analysis significantly advance the state of the art in CS. The encouraging nature of this initial analysis is meant to introduce the machine-learning community to a new problem for which it is poised to make an important contribution.

2. Compressive-Sensing Inversion From Bayesian Viewpoint

2.1. Compressive Sensing as Linear Regression

It was assumed at the start that $\boldsymbol{f}$ is compressible in the basis $\boldsymbol{B}$. 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

$$\boldsymbol{g} = \boldsymbol{\Phi w} = \boldsymbol{\Phi w}_s + \boldsymbol{\Phi w}_e = \boldsymbol{\Phi w}_s + \boldsymbol{n}_e, \qquad (2)$$

where $\boldsymbol{n}_e = \boldsymbol{\Phi w}_e$. 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 [Papoulis & Pillai, 2002] 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

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

where the components of $\boldsymbol{n}$ are approximated as a zero-mean Gaussian noise with unknown variance σ^2.

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 [Tipping, 2001, Wipf et al., 2004]. 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.

2.2. 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 [Figueiredo, 2002]:

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

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. Given the CS measurements $\boldsymbol{g}$, and assuming the model in (3), 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 (4).

3. Estimate of Full Posterior for Sparse Weights

3.1. 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 σ^2.

Rather than imposing a Laplace prior on $\boldsymbol{w}$, we develop a hierarchical prior [Tipping, 2001, Figueiredo, 2002] that has similar properties but that allows convenient conjugate-exponential analysis. Specifically, we introduce the prior

$$p(\boldsymbol{w}|\boldsymbol{\alpha}) = \prod_{i=1}^{N} \mathcal{N}(w_i|0, \alpha_i^{-1}), \qquad (5)$$

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

$$p(\boldsymbol{\alpha}|a, b) = \prod_{i=1}^{N} \Gamma(\alpha_i|a, b). \qquad (6)$$

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

$$p(\boldsymbol{w}|a, b) = \prod_{i=1}^{N} \int_0^\infty \mathcal{N}(w_i|0, \alpha_i^{-1}) \Gamma(\alpha_i|a, b) d\alpha_i. \qquad (7)$$

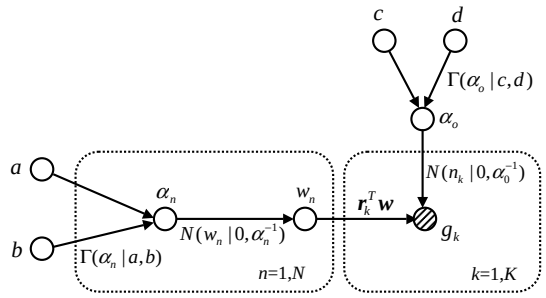

Figure 1. Graphical model of the Bayesian CS formulation.

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 [Tipping, 2001]. 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 (7) favors most w_i being zero (i.e., it is a sparseness prior). Similarly, we introduce the Gamma prior $\Gamma(\alpha_0|c,d)$ on the inverse of the noise variance $\alpha_0=1/\sigma^2$.

To see the advantage of the sparseness prior in (7), 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 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 $\boldsymbol{w}$ and α_0. For example, one may conveniently implement a Markov Chain Monte Carlo (MCMC) [Gilks et al., 1996] or, more efficiently and approximately, a variational Bayesian (VB) analysis [Bishop & Tipping, 2000]. While the VB analysis is efficient relative to MCMC, we here consider a type-II maximum-likelihood (ML) analysis, with the objective of achieving highly efficient computations while still preserving accurate results [Tipping, 2001].

3.2. The Bayesian CS algorithm

As shown by Tipping [2001], 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 Gaussian distribution with mean and covariance:

$$\boldsymbol{\mu} = \alpha_0 \boldsymbol{\Sigma} \boldsymbol{\Phi}^T \boldsymbol{g}, \tag{8}$$

$$\boldsymbol{\Sigma} = (\boldsymbol{A} + \alpha_0 \boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1}, \tag{9}$$

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

$$\mathcal{L}(\boldsymbol{\alpha},\alpha_0)=\log p(\boldsymbol{g}|\boldsymbol{\alpha},\alpha_0)=\log\int p(\boldsymbol{g}|\boldsymbol{w},\alpha_0)p(\boldsymbol{w}|\boldsymbol{\alpha})d\boldsymbol{w}$$

$$=-\frac{1}{2}\left[K\log 2\pi+\log|\boldsymbol{C}|+\boldsymbol{g}^T\boldsymbol{C}^{-1}\boldsymbol{g}\right], \tag{10}$$

with $\boldsymbol{C}=\sigma^2\boldsymbol{I}+\boldsymbol{\Phi}\boldsymbol{A}^{-1}\boldsymbol{\Phi}^T$. Then a type-II ML solution employs the point estimates for $\boldsymbol{\alpha}$ and α_0 that maximize (10). This can be implemented readily via the EM algorithm or direct differentiation [Tipping, 2001], to yield:

$$\alpha_i^{new}=\gamma_i/\mu_i^2, \qquad i\in\{1,2,...,N\}, \tag{11}$$

with $\gamma_i \triangleq 1-\alpha_i\boldsymbol{\Sigma}_{ii}$, where $\boldsymbol{\Sigma}_{ii}$ is the ith diagonal element from $\boldsymbol{\Sigma}$ in (9), and

$$1/\alpha_0^{new}=\frac{||\boldsymbol{g}-\boldsymbol{\Phi}\boldsymbol{\mu}||_2^2}{N-\sum_i\gamma_i}. \tag{12}$$

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 (8)-(9) and (11)-(12), 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 for representation of $\boldsymbol{g}$, and the level of sparseness (size of M) is determined automatically [Wipf et al., 2004]. It is also important to note that, as a result of the type-II ML solution, 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.

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 (8)-(9), then $\boldsymbol{f}$ is also drawn from a multivariate Gaussian distribution, with mean and covariance

$$E(\boldsymbol{f}) = \boldsymbol{B}\boldsymbol{\mu}, \tag{13}$$

$$Cov(\boldsymbol{f}) = \boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T. \tag{14}$$

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

While the iterative algorithm described above has been demonstrated to yield a highly accurate sparse linear-regression representation [Tipping, 2001], we note the following practical limitation. When evaluating (9) one must invert matrices of size $N \times N$, which has complexity $\mathcal{O}(N^3)$, thereby making this approach relatively slow for data $\boldsymbol{f}$ of large dimension N. This motivates development of a fast Bayesain algorithm with the objective of achieving highly efficient computations comparable to existing CS algorithms, such as OMP [Tropp & Gilbert, 2005] and StOMP [Donoho et al., 2006].

Fortunately, this fast Bayesian algorithm has been developed in [Faul & Tipping, 2002] by analyzing the properties of the marginal likelihood function in (10). This enables a principled and efficient sequential addition and deletion of candidate basis function (columns of $\boldsymbol{\Phi}$) to monotonically maximize the marginal likelihood. Because of limited space, we omit the detailed discussion of this fast algorithm, and refer the reader to [Tipping & Faul, 2003] for more information. We here only briefly summarize some of its key properties. Compared with the iterative algorithm presented above, the fast algorithm operates in a constructive manner, i.e., sequentially adds terms to the model until all M nonzero weights have been added. So, the complexity of the algorithm is more related to M than N. Further, by using the matrix inverse identity, the inverse operation in (9) has been implemented by an iterative update formula with reduced complexity. Detailed analysis shows that this fast algorithm has complexity $\mathcal{O}(NM^2)$, which is more efficient than the original RVM, especially when the underlying signal is truly sparse ($M \ll N$). Unlike other related CS algorithms (e.g., OMP and StOMP), the fast algorithm has the operation of deleting a basis function from the model (i.e., setting $\alpha_i = \infty$). This deletion operation is the likely explanation for the improvement in sparsity of this fast algorithm demonstrated in the experiments (see Sec. 5). More importantly, while conventional CS algorithms yield a point estimate for $\boldsymbol{f}$, the Bayesian analysis considered here also yields the error bars defined in (14).

4. Adaptive Compressive Sensing

4.1. 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 [Tsaig & Donoho, 2006]. In addition, previous CS algorithms [Donoho, 2006, Tropp & Gilbert, 2005, Donoho et al., 2006] 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 not feasible. The Bayesian CS (BCS) algorithm introduced in Sec. 3.2 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 studied previously in machine learning under the name of experiment design or active learning [Fedorov, 1972,MacKay, 1992]. 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 [Cover & Thomas, 1991] for $\boldsymbol{f}$ therefore satisfies:

$$\begin{aligned} h(\boldsymbol{f}) &= \frac{1}{2}\log|\boldsymbol{B}\boldsymbol{\Sigma}\boldsymbol{B}^T| + c = \frac{1}{2}\log|\boldsymbol{\Sigma}| + c \\ &= -\frac{1}{2}\log|\boldsymbol{A} + \alpha_0\boldsymbol{\Phi}^T\boldsymbol{\Phi}| + c, \end{aligned} \tag{15}$$

where c is a constant, independent of $\boldsymbol{\Phi}$. Recall that $\boldsymbol{A} = \mathrm{diag}(\alpha_1, \dots, \alpha_N)$, 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. 3.2).

We may now ask which new projection $\boldsymbol{r}_{K+1}$ would be optimal for minimizing the differential entropy in (15). 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

$$h_{new}(\boldsymbol{f}) = h(\boldsymbol{f}) - \frac{1}{2}\log\left[1 + \alpha_0\boldsymbol{r}_{K+1}^T\boldsymbol{\Sigma}\,\boldsymbol{r}_{K+1}\right], \tag{16}$$

where α_0 and $\boldsymbol{\Sigma}$ are based on estimates found using the previous K projections. In order to minimize (16) the next projection $\boldsymbol{r}_{K+1}$ must be designed to maximize the variance of the *expected* measurement g_{K+1} since

$$\boldsymbol{r}_{K+1}^T\boldsymbol{\Sigma}\,\boldsymbol{r}_{K+1} = \boldsymbol{r}_{K+1}^T\mathrm{Cov}(\boldsymbol{w})\,\boldsymbol{r}_{K+1} = \mathrm{Var}(g_{K+1}). \tag{17}$$

In other words, $\boldsymbol{r}_{K+1}$ must 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 adaptive algorithm 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 $\mathbf{L}$ of possible next projections, with $\boldsymbol{r}_{K+1}$ selected from $\mathbf{L}$ with the goal of maximizing (17).

An additional issue needs to be clarified if the eigenvector of $\boldsymbol{\Sigma}$ is used for 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 Gaussian 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 $||\boldsymbol{r}_{K+1}||_2 = 1.01$.

4.2. Approximate Adaptive CS

The error bars on the estimate of $\boldsymbol{f}$ play a critical role in implementing the above adaptive CS scheme, with this 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 (16) relies 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 (12) we have $\sigma^2 = 1/\alpha_0 = ||\boldsymbol{g} - \boldsymbol{\Phi}\boldsymbol{\mu}||_2^2/(N - \sum_i \gamma_i)$, 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 a CS algorithm such as OMP or StOMP 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. 3.2 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. 5, when presenting example results, we make comparisons between the rigorous implementation discussed in Sec. 4.1 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.

5. Example Results

We test the performance of BCS on several example problems considered widely in the CS literature, with comparisons made to Basis Pursuit (BP) [Chen et al., 1999] and StOMP [Donoho et al., 2006]. While BP is a relatively computationally expensive algorithm that involves linear programming, StOMP may be one of the state-of-the-art fast CS algorithms. In the experiments we evaluate the reconstruction error as $||\boldsymbol{f}_{method} - \boldsymbol{f}||_2/||\boldsymbol{f}||_2$. All the computations presented here were performed using Matlab run on a 3.4GHz Pentium machine.

5.1. BCS and Projection Optimization

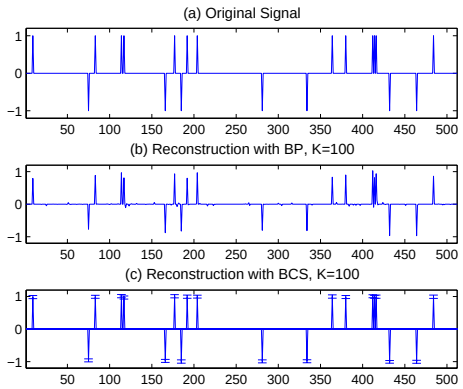

Figure 2. Reconstruction of *Spikes*. (a) Original signal; (b) Reconstruction with BP, err_{BP}=0.158, t_{BP}=1.56 secs; (c) Reconstruction with BCS, err_{BCS}=0.015, t_{BCS}=0.63 secs.

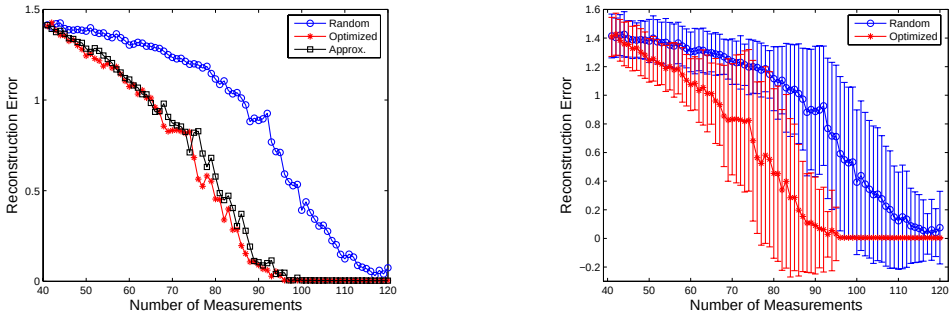

Figure 3. Comparison of adaptive and random projections, with the first 40 projections performed randomly. (a) Reconstruction error of BCS with random projections, optimized projections (Sec. 4.1) and approximated projections (Sec. 4.2); the results are averaged over 100 runs; (b) the variances of the reconstruction error of BCS with random projections and optimized projections (Sec. 4.1); the variance for the approximate scheme in Sec. 4.2 is very similar to that for Sec. 4.1, and thus is omitted to improve visibility.

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/`.

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 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. 4, 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. 4.1. 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. 4.2, 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. 4.2 yields results very comparable to the more-rigorous analysis in Sec. 4.1. 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.

5.2. BCS vs. BP and StOMP

In the following set of experiments, the performance of BCS is compared to 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 $\mathbf{\Phi}$ here are drawn from a uniform spherical distribution [Tsaig & Donoho, 2006].

5.2.1. Random-Bars

Figure 4 shows the reconstruction results for *Random-Bars* that has been used in [Tsaig & Donoho, 2006].

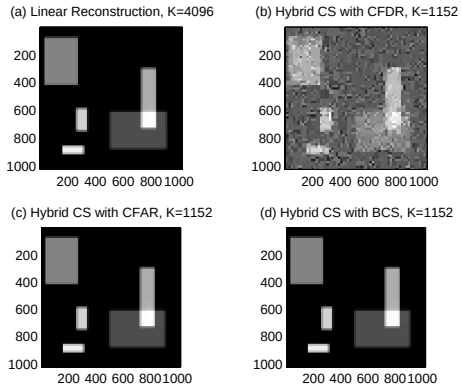

Figure 4. Reconstruction of *Random-Bars* with hybrid CS. (a) Linear reconstruction from K=4096 samples, err_{LIN}= 0.2271; (b) Reconstruction with CFDR, err_{CFDR}=0.5619, t_{CFDR}=4 secs; (c) Reconstruction with CFAR, err_{CFAR}= 0.2271, $t_{CFAR} = 5$ secs; (c) Reconstruction with BCS, $err_{BCS} = 0.2271$, $t_{BCS} = 15$ secs. BP took 108 secs with the reconstruction error 0.2279, which is not shown here.

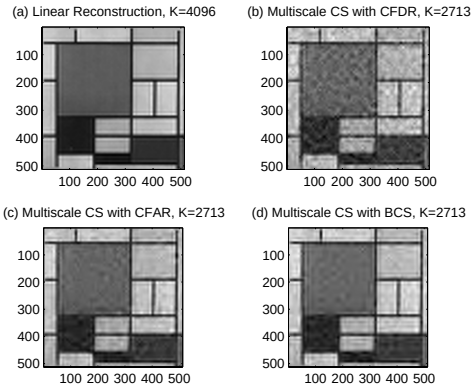

Figure 5. Reconstruction of *Mondrian* with multiscale CS. (a) Linear reconstruction from K=4096 samples, err_{LIN}= 0.1333; (b) Reconstruction with CFDR, err_{CFDR}=0.1826, t_{CFDR}=10 secs; (c) Reconstruction with CFAR, err_{CFAR}= 0.1508, $t_{CFAR} = 28$ secs; (c) Reconstruction with BCS, $err_{BCS} = 0.1503$, $t_{BCS} = 18$ secs. BP took 162 secs with the reconstruction error 0.1416, which is not shown here.

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 4(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. 4(b-d) have results for the hybrid CS scheme [Tsaig & Donoho, 2006] with $K = 1152$ 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. 4(b).

5.2.2. Mondrian

Figure 5 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 [Tsaig & Donoho, 2006] for image reconstruction, with a coarsest scale $j_0 = 4$, and a finest scale $j_1 = 6$ on the "symmlet8" wavelet. Figure 5(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. 5(b-d) have results for the multiscale CS scheme with K=2713 multiscale compressed samples. In the example results in Figs. 5(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).

Table 1. Summary of the performances of BP, StOMP and BCS on Mondrian.

	BP	CFDR	CFAR	BCS
# Nonzeros	3840	1766	926	615
Time (secs)	162	10	28	18
Reconst. Error	0.1416	0.1826	0.1508	0.1503

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 1. Evidently, BCS found the sparsest solution (with 615 nonzeros) relative to the two StOMP implementations, but yielded the second smallest reconstruction error (0.1503). This indicates that although each iteration of StOMP allows multiple nonzero weights to be added into the "active set" [Donoho et al., 2006], 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.

6. Conclusions

Compressive sensing has been considered from a Bayesian perspective. It has been demonstrated that by utilizing the previously derived algorithms (relevance vector machine) from the sparse Bayesian 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 [Chen et al., 1999, Donoho et al., 2006]. On the examples considered from the literature, the BCS solution typically has computation time comparable to state-of-the-art algorithms such as StOMP [Donoho et al., 2006]; 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 $\mathbf{\Phi}$. Experiments on synthetic data demonstrate a significantly accelerated rate of convergence compared to the original CS construction. Finally, a simple approximate scheme has been introduced, which allows existing CS algorithms to benefit immediately from this Bayesian analysis.

There is a clear connection between CS and regression shrinkage and selection via the Lasso [Tibshirani, 1996, Efron et al., 2004] as both focus on solving the objective function (1). Research on Lasso has produced algorithms that might have some relevance to CS and BCS. In addition to this, other possible areas of future research may include simultaneous inversion of multiple data sets, borrowing ideas from multi-task learning [Caruana, 1997], and a theoretical analysis of adaptive CS, which can be an important complement to the existing analysis for the conventional CS formulation [Haupt & Nowak, 2006].

Acknowledgments

The authors wish to thank Ya Xue for the initial discussions and suggestions on compressive sensing, and the anonymous reviewers for the constructive suggestions. The authors also thank E. Candès, J. Romberg and D. Donoho *et al.* for sharing the ℓ_1-Magic and SparseLab online. Their generous distribution of the code made the experimental comparisons in this paper very convenient.

References

Bishop, C. M., & Tipping, M. E. (2000). Variational relevance vector machines. *UAI 16* (pp. 46–53).

Candès, E., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Information Theory*, *52*, 489–509.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*, 41–75.

Chen, S., Donoho, D. L., & Saunders, M. A. (1999). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, *20*, 33–61.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York, NY: Wiley.

Donoho, D. L. (2006). Compressed sensing. *IEEE Trans. Information Theory*, *52*, 1289–1306.

Donoho, D. L., Tsaig, Y., Drori, I., & Starck, J.-C. (2006). Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. Preprint.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics*, *32*, 407–499.

Faul, A. C., & Tipping, M. E. (2002). Analysis of sparse Bayesian learning. *NIPS 14*.

Fedorov, V. V. (1972). *Theory of optimal experiments*. Academic Press.

Figueiredo, M. (2002). Adaptive sparseness using Jeffreys prior. *NIPS 14)*.

Gilks, W. R., Richardson, S., & Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall.

Haupt, J., & Nowak, R. (2006). Signal reconstruction from noisy random projections. *IEEE Trans. Information Theory*, *52*, 4036–4048.

MacKay, D. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*, 590–604.

Mallat, S. (1998). *A wavelet tour of signal processing*. Academic Press. 2nd edition.

Papoulis, A., & Pillai, S. U. (2002). *Probability, random variables and stochastic processes*. McGraw-Hill. 4th edition.

Pearlman, W. A., Islam, A., Nagaraj, N., & Said, A. (2004). Efficient, low-complexity image coding with a set-partitioning embedded block coder. *IEEE Trans. Circuits Systems Video Technology*, *14*, 1219–1235.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, *58*, 267–288.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, *1*, 211–244.

Tipping, M. E., & Faul, A. C. (2003). Fast marginal likelihood maximisation for sparse Bayesian models. *Proc. of the 9th International Workshop on AIStats*.

Tropp, J. A., & Gilbert, A. C. (2005). Signal recovery from partial information via orthogonal matching pursuit. Preprint.

Tsaig, Y., & Donoho, D. L. (2006). Extensions of compressed sensing. *Signal Processing*, *86*, 549–571.

Wipf, D., Palmer, J., & Rao, B. (2004). Perspectives on sparse Bayesian learning. *NIPS 16*.