Multi-Task Compressive Sensing

Shihao Ji, David Dunson†, and Lawrence Carin
Department of Electrical and Computer Engineering
†Institute of Statistics and Decision Sciences
Duke University, Durham, NC 27708-0291 USA
{shji,lcarin}@ece.duke.edu, dunson@stat.duke.edu

Abstract

Compressive sensing (CS) is a framework whereby one performs *n non-adaptive* measurements to constitute an n-dimensional vector v, with v used to recover an *m*-dimensional approximation $\hat{u}$ to a desired *m*-dimensional signal u, with $n \ll m$; this is performed under the assumption that u is sparse in the basis represented by the matrix Ψ, the columns of which define discrete basis vectors. It has been demonstrated that with appropriate design of the compressive measurements used to define v, the decompressive mapping $v \rightarrow \hat{u}$ may be performed with error $\|u-\hat{u}\|_2^2$ having asymptotic properties (large n and $m > n$) analogous to those of the best *adaptive* transform-coding algorithm applied in the basis Ψ. The mapping $v \rightarrow \hat{u}$ constitutes an inverse problem, often solved using ℓ_1 regularization or related techniques. In most previous research, if multiple compressive measurements $\{v_i\}_{i=1,M}$ are performed, each of the associated $\{\hat{u}_i\}_{i=1,M}$ are recovered one at a time, independently. In many applications the *M* "tasks" defined by the mappings $v_i \rightarrow \hat{u}_i$ are not statistically independent, and it may be possible to improve the performance of the inversion if statistical inter-relationships are exploited. In this paper we address this problem within a multi-task learning setting, wherein the mapping $v_i \rightarrow \hat{u}_i$ for each task corresponds to inferring the parameters (here, wavelet coefficients) associated with the desired signal u_i, and a shared prior is placed across all of the *M* tasks. In this multi-task learning framework data from all M tasks contribute toward inferring a posterior on the hyperparameters, and once the shared prior is thereby inferred, the data from each of the M individual tasks is then employed to estimate the task-dependent wavelet coefficients. An empirical Bayes procedure and fast inference algorithm is developed. Example results are presented on several data sets.

Index Terms

Compressive sensing (CS), Multi-task learning, Sparse Bayesian learning, Hierarchical Bayesian modeling, Modified relevance vector machine

I. INTRODUCTION

The development of wavelets [1], [2] has had a significant impact on several areas of signal processing and compression. An important characteristic of wavelets is the sparse representation of most natural signals in terms of a wavelet basis. Specifically, let u represent an *m*-dimensional real signal, let the $m \times m$ matrix Ψ denote a wavelet basis, and let the *m*-dimensional vector θ represent the wavelet coefficients, and hence $u = \Psi\theta$. We further let $u_N = \Psi\theta_N$ represent an approximation to u, where θ_N is the same as θ except that the $m - N$ smallest coefficients are set to zero. The compressive properties of wavelets assures that $\|u - u_N\|_2^2$ is typically small for $N \ll m$, thereby motivating the use of wavelets in a new generation of compression techniques for images and video [3], [4].

While wavelets have had a profound impact on practical compression schemes, there are issues that warrant further investigation. For example, while most natural signals are highly compressible in a wavelet basis, the specific N wavelet coefficients that have largest amplitude varies strongly from signal to signal. The aforementioned compression techniques must therefore adapt to each new signal under test, this constituting the principal complexity of wavelet-based compression algorithms. Of more practical importance, while the approximated signal u_N is highly compressed ($N \ll m$), one first had to measure the m-dimensional signal u, and in some sense $m - N$ pieces of data were measured unnecessarily. This latter issue raises the following question: Is it possible to measure the informative part of the signal directly, such that most unnecessary measurements are avoided from the start? This question has been answered in the affirmative, with this spawning the new field of compressive sensing (CS) [5], [6].

When performing compressive measurements, one does not attempt to directly measure the N dominant wavelet coefficients, as this would require adapting to each new signal. Rather, in a CS measurement one implicitly measures *all* of the wavelet coefficients, with these measurements performed by projecting the signal of interest u on an m-dimensional vector with *random* weights [5], [6]. Each set of random weights (*i.e.,* each random projection) corresponds to one CS measurement, and n such measurements constitute the overall CS measurement vector v. The CS measurement may be expressed in matrix form as $v = \Phi\Psi^T u$, where Φ is an $n \times m$ matrix, with each component a draw from a random variable; each row of the matrix product $\Phi\Psi^T$ corresponds to one of the n random vectors on which the signal of interest u is projected. The mapping from the CS data v to an approximation of the underlying signal u, with the approximation represented as $\hat{u}$, is under-determined, because $n \ll m$. However, by exploiting the fact that u is compressible in the basis Ψ, and using a properly designed matrix Φ, in the asymptotic limit (large n, and $m > n$) the following property holds with high probability: the optimal $\hat{u}$ estimated

from the *non-adaptive* CS measurements v has error $\|u - \hat{u}\|_2^2$ proportional to that of $\|u - u_N\|_2^2$; the aforementioned probability is controlled by the size of n [5], [6]. This is a remarkable result, because to constitute u_N one first must measure the m-dimensional signal u, represent it in the basis Ψ, and then *adaptively* determine u_N, while in the CS measurement the n-dimensional signal v is measured *non-adaptively*, and we may employ $n \ll m$ CS measurements if u is highly compressible in the basis Ψ. The utility of this framework has motivated development over the last few years of several techniques for performing the CS inversion $v \rightarrow \hat{u}$ [7]–[11].

Before proceeding, it should be emphasized that CS is a framework that is not limited to wavelet-based representations. While wavelets played a key role in early developments of sparse signal representations and compression, CS is applicable to any signal representation for which most of the basis-function coefficients are small (implying a sparse representation). Therefore, for particular classes of signals a Fourier, Gabor, or other representation may be appropriate. Wavelets represent an important basis within the context of CS, but by no means the only one. In addition, Ψ may not be a basis, for a compressible frame also meets the CS conditions [6]. Therefore, we underscore that CS is a highly flexible framework.

While there have been numerous techniques developed to constitute the inverse CS mapping $v \rightarrow \hat{u}$, typically these algorithms perform the inversion separately and independently for each compressive measurement v. In practice one may perform multiple CS measurements, and M such measurements are denoted $\{v_i\}_{i=1,M}$. One may anticipate that many of the measurements in the set $\{v_i\}_{i=1,M}$ are statistically related, particularly when repeated measurements are taken of the same scene or for the same type of diagnostic task (*e.g.*, repeated MRI images performed in a CS setting). By exploiting the statistical relationships between these M measurements, one may hope to constitute the M mappings $v_i \rightarrow \hat{u}_i$ with fewer total measurements. Specifically, if n_i CS measurements are performed for the ith task to perform the *independent* mapping $v_i \rightarrow \hat{u}_i$ with desired accuracy, then ideally less than $\sum_{i=1}^{M} n_i$ total CS measurements would be required by exploiting statistical inter-relationships between the M sensing "tasks".

The mapping $v_i \rightarrow \hat{u}_i$ may be framed as a regression problem [10]. Specifically, assume that the desired signal $u_i = \Psi\theta_i$, where $u_i \in \mathbb{R}^m$ is a column vector, Ψ is again a matrix of size $m \times m$ representing the basis of interest (*e.g.*, wavelets), and $\theta_i \in \mathbb{R}^m$ is a column vector of corresponding basis-function coefficients. Compressive sensing exploits the fact that we know *a priori* that most components of θ_i can be ignored, and hence that u_i may be represented sparsely in the basis Ψ. This may therefore be framed as a sparse linear-regression problem, assuming the linear relationship between the $n_i \ll m$ CS measurements v_i and the desired u_i is known, with this represented in terms of the $n_i \times m$ dimensional

matrix Φ_i: $v_i = \Phi_i \Psi^T u_i = \Phi_i \theta_i$. The sparse regression problem is constituted in terms of solving for θ_i such that $v_i = \Phi_i \theta_i$, under the constraint that θ_i is sparse (*i.e.,* most components of θ_i vanish, or at least may be set to zero with minimal impact on the reconstruction of u_i). This relationship to linear regression makes existing sparse regression algorithms particularly relevant for CS inversion, for example [12]–[15].

Each of the CS measurements $\{v_i\}_{i=1,M}$ yields a corresponding regression "task" $v_i \rightarrow \hat{\theta}_i$, and performing multiple such learning tasks has been referred to in the machine-learning community as multi-task learning [16]; $\hat{u}_i$ satisfies $\hat{u}_i = \Psi \hat{\theta}_i$. Typical approaches to information transfer among tasks include: sharing hidden nodes in neural networks [16]–[18], placing a common prior in hierarchical Bayesian models [19]–[22], sharing parameters of Gaussian processes [23], sharing a common structure on the predictor space [24], and structured regularization in kernel methods [25], among others.

In statistics, the problem of combining information from similar but independent experiments has been studied in the field of meta-analysis [26] for a variety of applications in medicine, psychology and education. Hierarchical Bayesian modeling is one of the most important methods for meta analysis [27]–[31]. Hierarchical Bayesian models provide the flexibility to model both the individuality of tasks (experiments), and the correlations between tasks. Statisticians refer to this approach as "borrowing strength" across tasks. Usually the bottom layer of the hierarchy is composed of individual models with task-specific parameters. On the layer above, tasks are connected together via a common prior placed on those parameters; on a layer above is a hyper-prior, invoked on parameters of the prior at the level below. The hierarchical model can achieve efficient information-sharing between tasks for the following reason. Learning of the common prior is also a part of the training process, and data from all tasks contribute to learning the common prior, thus making it possible to transfer information between tasks (via sufficient statistics). Given the prior, individual models are learned independently. As a result, the estimation of a regressor (task) is affected by both its own training data and by data from the other tasks related through the common prior.

In the work presented here we process the data $\{v_i\}_{i=1,M}$ jointly, simultaneously implementing the mapping $v_i \rightarrow \hat{\theta}_i$ on all M regression tasks, where we emphasize that the goal is to estimate the sparse wavelet coefficients θ_i, from which u_i is inferred. Borrowing ideas from the aforementioned hierarchical models, a common parametric prior is placed on all coefficients $\{\theta_i\}_{i=1,M}$, and a sparseness-promoting hyperprior is also invoked. The data from all CS measurements $\{v_i\}_{i=1,M}$ are employed to jointly learn a posterior density function on the parameters of the prior, and this shared updated prior is then used to perform inference on the parameters θ_i associated with each CS task. In this manner data from all CS

tasks are learned to update the parameters of the prior, and then the task-specific CS data v_i is used to provide inference on the associated parameters θ_i, with the corresponding estimate denoted $\hat{\theta}_i$.

While the formulation is constituted in a fully Bayesian setting, hyper-parameters are estimated using an empirical Bayes procedure. This yields a computationally efficient multi-task CS inference algorithm, that extends previous research in Bayesian CS analysis [10]. This paper therefore extends the work presented in [10], wherein the Bayesian inversion $v_i \rightarrow \hat{\theta}_i$ was performed one task at a time.

Besides many other advantages of Bayesian analysis of CS (i.e., measures of uncertainty, adaptive design of projection, etc. [10]), (hierarchical) Bayesian analysis also provides a flexible framework for multi-task CS. Conventional CS inverse algorithms [7]–[9] typically employ a point estimate for θ_i, and therefore are not directly amenable for information transfer among related multiple CS tasks.

In addition to developing a multi-task CS framework, a modified sparse regression model is introduced, of interest even for the traditional single-task CS setting. As discussed further below, this analysis analytically integrates out the noise-variance term in the regression model, and it is demonstrated that this yields improved algorithmic performance.

The remainder of the paper is organized as follows. In Sec. II we consider the multi-task CS inversion problem from a hierarchical Bayesian perspective, and make connections with what has been done previously for this problem; a fast algorithm is developed for inference for multi-task CS. The model in Sec. II builds naturally upon previous sparse-regression analyses of this type. In Sec. III we introduce a modified sparse-regression model, through analytic integrating out of the noise variance in the regression model, as well as a fast inference algorithm. Example results on multiple datasets are presented in Sec. IV, with comparisons to single-task CS inversion. Conclusions and future work are discussed in Sec. V.

II. Hierarchical Multi-Task CS Modeling

A. Bayesian Regression Formulation

Assume that M CS measurements are performed, with these multiple sensing tasks statistically inter-related, as defined precisely below. The M measurements are represented as $\{v_i\}_{i=1,M}$, where $v_i = \Phi_i \Psi^T u_i = \Phi_i \theta_i$ where in general the M measurements employ different $n_i \times m$ random CS projection matrices $\{\Phi_i\}_{i=1,M}$. In the context of a regression analysis, we assume [10]

$$v_i = \Phi_i \theta_i + \nu_i, \tag{1}$$

where $\nu_i \in \mathbb{R}^{n_i}$ is a residual error vector, modeled as n_i i.i.d. draws of a zero-mean Gaussian random variable with unknown precision α_0 (variance $1/\alpha_0$). The likelihood function for the parameters θ_i and

α_0, based on the observed data v_i, may therefore be expressed as

$$p(v_i|\theta_i, \alpha_0) = (2\pi/\alpha_0)^{-n_i/2} \exp\{-\frac{\alpha_0}{2}\|v_i - \Phi_i\theta_i\|_2^2\}. \quad (2)$$

The parameters θ_i (here, wavelet coefficients) characteristic of task i are assumed to be drawn from a common zero-mean Gaussian distribution, and it is in this sense that the M tasks are statistically related. Specifically, letting $\theta_{i,j}$ represent the jth wavelet (or scaling function) coefficient for CS task i, we have

$$p(\theta_i|\alpha) = \prod_{j=1}^{m} \mathcal{N}(\theta_{i,j}|0, \alpha_j^{-1}). \quad (3)$$

It is important to note that the hyper-parameters $\alpha = \{\alpha_j\}_{j=1,m}$ are shared among all M tasks, and therefore the data from all CS measurements, $\{v_i\}_{i=1,M}$ will contribute to learning the hyper-parameters, offering the opportunity to adaptively borrow strength from the different measurements to a degree controlled by α.

Gamma priors are placed on the parameters $\{\alpha_j\}_{j=1,m}$ and α_0. Specifically,

$$p(\alpha|c, d) = \prod_{j=1}^{m} Ga(\alpha_j|c, d) = \prod_{j=1}^{m} \frac{d^c}{\Gamma(c)} \alpha_j^{(c-1)} \exp(-d\alpha_j). \quad (4)$$

It has been demonstrated [12] that appropriate choice of parameters c and d encourages a sparse representation for the coefficients in the vector θ_i, where here this concept is extended to a multi-task CS setting. We find that $c = d = \epsilon$, with $\epsilon > 0$ a small constant, leads to procedures with good performance. In this case, $Ga(\cdot|c, d)$ has a large spike concentrated at zero and a heavy right tail. The spike corresponds to basis functions for which there is essentially no borrowing of information. Such basis functions characterize components that are idiosyncratic to specific signals. At the other extreme, basis functions for which α_j is in the right tail have coefficients that are shrunk strongly to zero for all tasks, favoring spareness, while borrowing information about which basis functions are not important for any of the signals in the collection. For small ϵ, there will be many such basis functions. As a default choice which avoids subjective choice of c, d and leads to computational simplifications, we recommend letting $c = d = 0$.

We similarly define a Gamma prior on the noise precision

$$p(\alpha_0|a, b) = Ga(\alpha_0|a, b). \quad (5)$$

For this prior, we again recommend $a = b = 0$ as a default choice. For α_0, this choice corresponds to a commonly-used improper prior expressing *a priori* ignorance about plausible values for the residual precision.

Given the observed CS data $\{v_i\}_{i=1,M}$ from the (assumed) statistically related data, one may in principle infer a posterior density function on the hyper-parameters α and the noise precision α_0,

$$p(\alpha,\alpha_0|\{v_i\}_{i=1,M},a,b,c,d)=\frac{p(\alpha_0|a,b)p(\alpha|c,d)\prod_{i=1}^{M}\int d\theta_i p(v_i|\theta_i,\alpha_0)p(\theta_i|\alpha)}{\int d\alpha\int d\alpha_0 p(\alpha_0|a,b)p(\alpha|c,d)\prod_{i=1}^{M}\int d\theta_i p(v_i|\theta_i,\alpha_0)p(\theta_i|\alpha)}. \quad (6)$$

We note that the integral in (6) with respect to α is actually an m-dimensional integral, with each integral linked to one component of α; similarly, each integral with respect to θ_i is an m-dimensional integral, over all wavelet-coefficient weights. To avoid the complexity of evaluating some of these integrals, particularly those with respect to α and α_0, we seek a point estimate for the parameters α and α_0, and a maximum *a posterior* (MAP) estimate for α and α_0 is found as

$$\{\alpha^{MAP},\alpha_0^{MAP}\}=\arg\max_{\alpha,\alpha_0}\{\ \log p(\alpha_0|a,b)+\log p(\alpha|c,d)+\sum_{i=1}^{M}\log\int d\theta_i\ p(v_i|\theta_i,\alpha_0)p(\theta_i|\alpha)\ \}, \quad (7)$$

which reduces to the simplified form in the limit as $a,b,c,d\rightarrow 0$:

$$\{\alpha^{ML},\alpha_0^{ML}\}=\arg\max_{\alpha,\alpha_0}\sum_{i=1}^{M}\log\int d\theta_i\ p(v_i|\theta_i,\alpha_0)p(\theta_i|\alpha), \quad (8)$$

which can be interpreted as a MAP estimate under an improper, default prior or as a maximum likelihood (ML) estimate.

Let (α^P,α_0^P) represent point estimates for α and α_0, based on either of the MAP approximations. Using (α^P,α_0^P), we may analytically evaluate the posterior density function for the basis-function coefficients θ_i. In particular, using (2) and (3), and the point estimate (α^P,α_0^P), we have

$$p(\theta_i|v_i,\alpha^P,\alpha_0^P)=\frac{p(v_i|\theta_i,\alpha_0^P)p(\theta_i|\alpha^P)}{\int d\theta_i p(v_i|\theta_i,\alpha_0^P)p(\theta_i|\alpha^P)}. \quad (9)$$

The expression in (9) may be evaluated in closed form [12] to yield

$$p(\theta_i|v_i,\alpha^P,\alpha_0^P)=\mathcal{N}(\theta_i|\mu_i,\Sigma_i), \quad (10)$$

with

$$\mu_i = \alpha_0^P\Sigma_i\Phi_i^T v_i, \quad (11)$$

$$\Sigma_i = (\alpha_0^P\Phi_i^T\Phi_i+\mathrm{A})^{-1}, \quad (12)$$

where $\mathrm{A}=\mathrm{diag}(\alpha_1^P,\alpha_2^P,\ldots,\alpha_m^P)$, and $(\alpha_1^P,\alpha_2^P,\ldots,\alpha_m^P)$ are the components of the vector α^P.

Before proceeding, we note the characteristics of the aforementioned algorithm. Using a MAP proce-

dure, one constitutes point estimates for the hyper-parameters α and α_0. Importantly, as implemented in (7) and (8), the hyper-parameter point estimates are based upon *all* of the observed CS measurements $\{v_i\}_{i=1,M}$, emphasizing the multi-task nature of the analysis (see Fig. 1). Subsequently, using the point estimate constituted using all of the data, a full posterior estimate is constituted for the basis-function coefficients $\{\theta_i\}_{i=1,M}$, where for this latter calculation θ_i is only dependent on v_i. Thus, to estimate the hyper-parameters all of the data are used, while to update an approximation to the wavelet coefficients θ_i only the associated task-dependent CS measurements are employed. This suggests an iterative algorithm that alternates between these global and local solutions, as outlined next.

This framework is related to extensive research in statistics on *empirical Bayesian analysis* [32]. Specifically, all of the data $\{v_i\}_{i=1,M}$ are used to constitute point estimates for the parameters α and α_0. Using the point estimate for α, one may specify the prior on the weights θ_i, via (3). Using this data-dependent (and hence "empirical") prior, the posterior on θ_i is evaluated analytically, as in (10). Again, all of the CS data $\{v_i\}_{i=1,M}$ are used to constitute the empirical Bayes prior, and then this prior is applied individually to each CS measurement v_i, to update the associated approximation to the parameters θ_i, and the algorithm iterates between these two steps.

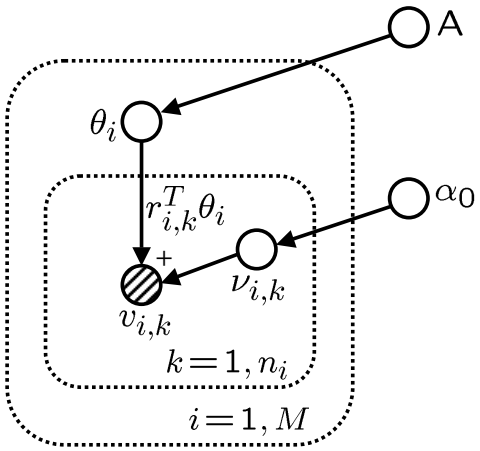

Fig. 1. A hierarchical Bayesian model representation of the mutli-task CS, where $\Phi_i = [r_1, r_2, \ldots, r_{n_i}]^T$.

B. ML estimate for α and α_0

As discussed further when presenting results, we have found in practice that the MAP estimate in (8) (or equivalently the ML estimate) yields excellent results in practice, and the update equations are simpler than in the more general case (7). In addition, by choosing $a, b, c, d \rightarrow 0$, we obtain a default procedure

that avoids subjective choice of these key hyperparameters. To avoid confusion with the general MAP estimate in (7), we refer to (8) as the ML estimate through the remainder of the article. The ML estimate for α and α_0 is determined by maximizing the marginal log likelihood function

$$\begin{aligned}\mathcal{L}(\alpha,\alpha_0) &= \sum_{i=1}^{M}\log p(v_i|\alpha,\alpha_0)=\sum_{i=1}^{M}\log\int p(v_i|\theta_i,\alpha_0)p(\theta_i|\alpha)d\theta_i \\ &= -\frac{1}{2}\sum_{i=1}^{M}\left[n_i\log 2\pi+\log|\mathrm{C}_i|+v_i^T\mathrm{C}_i^{-1}v_i\right], \end{aligned} \tag{13}$$

with

$$\mathrm{C}_i=\alpha_0^{-1}\mathrm{I}+\Phi_i\mathrm{A}^{-1}\Phi_i^T. \tag{14}$$

1) Iterative Solution: Differentiating (13) with respect to α and α_0, setting the result to 0 followed by algebra, yields

$$\alpha_j^{new} = \frac{M}{\sum_{i=1}^{M}\mu_{i,j}^2}, \qquad j\in\{1,2,\ldots,m\}, \tag{15}$$

$$\alpha_0^{new} = \frac{\sum_{i=1}^{M}n_i}{\sum_{i=1}^{M}\|v_i-\Phi_i\mu_i\|^2}, \tag{16}$$

where $\mu_{i,j}$ is the jth component of μ_i.

2) Fast Algorithm: Similar to [33], considering the dependence of $\mathcal{L}(\alpha,\alpha_0)$ on a single hyperparameter α_j, $j\in\{1,2,\ldots,m\}$, we can decompose C_i in (14) as

$$\begin{aligned}\mathrm{C}_i &= \alpha_0^{-1}\mathrm{I}+\sum_{k\neq j}\alpha_k^{-1}\Phi_{i,k}\Phi_{i,k}^T+\alpha_j^{-1}\Phi_{i,j}\Phi_{i,j}^T \\ &= \mathrm{C}_{i,-j}+\alpha_j^{-1}\Phi_{i,j}\Phi_{i,j}^T, \end{aligned} \tag{17}$$

where $\Phi_i=[\Phi_{i,1},\Phi_{i,2},\ldots,\Phi_{i,m}]$, and $\mathrm{C}_{i,-j}$ is C_i with the contribution of basis function $\Phi_{i,j}$ removed. Applying matrix determinant and inverse identities, we can write the terms of interest in $\mathcal{L}(\alpha,\alpha_0)$ as

$$|\mathrm{C}_i|=|\mathrm{C}_{i,-j}|\,|1+\alpha_j^{-1}\Phi_{i,j}^T\mathrm{C}_{i,-j}^{-1}\Phi_{i,j}|, \tag{18}$$

$$\mathrm{C}_i^{-1}=\mathrm{C}_{i,-j}^{-1}-\frac{\mathrm{C}_{i,-j}^{-1}\Phi_{i,j}\Phi_{i,j}^T\mathrm{C}_{i,-j}^{-1}}{\alpha_j+\Phi_{i,j}^T\mathrm{C}_{i,-j}^{-1}\Phi_{i,j}}. \tag{19}$$

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

$$\begin{aligned}\mathcal{L}(\alpha, \alpha_0) &= -\frac{1}{2}\sum_{i=1}^{M}\left[n_i \log 2\pi + \log|\mathrm{C}_{i,-j}| + v_i^T \mathrm{C}_{i,-j}^{-1} v_i - \log\left(\frac{\alpha_j}{\alpha_j + s_{i,j}}\right) - \frac{q_{i,j}^2}{\alpha_j + s_{i,j}}\right] \\ &= \mathcal{L}(\alpha_{-j}, \alpha_0) + \frac{1}{2}\sum_{i=1}^{M}\left[\log\left(\frac{\alpha_j}{\alpha_j + s_{i,j}}\right) + \frac{q_{i,j}^2}{\alpha_j + s_{i,j}}\right] \\ &= \mathcal{L}(\alpha_{-j}, \alpha_0) + \ell(\alpha_j, \alpha_0), \end{aligned} \tag{20}$$

where α_{-j} is the same as α except the jth component is removed, and we have defined

$$s_{i,j} \triangleq \Phi_{i,j}^T \mathrm{C}_{i,-j}^{-1} \Phi_{i,j}, \quad \text{and} \quad q_{i,j} \triangleq \Phi_{i,j}^T \mathrm{C}_{i,-j}^{-1} v_i. \tag{21}$$

Differentiating $\ell(\alpha_j, \alpha_0)$ with respect to α_j and setting the result to zero, followed by algebra, yields

$$\sum_{i=1}^{M} \frac{s_{i,j}^2/\alpha_j + s_{i,j} - q_{i,j}^2}{(\alpha_j + s_{i,j})^2} = 0. \tag{22}$$

Except for the trivial solution $\alpha_j = \infty$, the other solutions of (22) cannot be expressed analytically. We thus assume that $\alpha_j \ll s_{i,j}$ (this has generally been found to be true numerically or causes more iterations upon convergence) and the denominator of (22) is now relatively invariant with respect to α_j. Therefore, we may approximate another solution as

$$\alpha_j \approx \frac{M}{\sum_{i=1}^{M}(q_{i,j}^2 - s_{i,j})/s_{i,j}^2}. \tag{23}$$

Immediately, we may recognize that (23) recovers the exact formula of α_j when $M = 1$ (single-task learning, as considered in [33]). Analysis of $\ell(\alpha_j, \alpha_0)$ when $M = 1$ shows that $\mathcal{L}(\alpha, \alpha_0)$ has a unique maximum with respect to α_j [34]:

$$\alpha_j \approx \frac{M}{\sum_{i=1}^{M}(q_{i,j}^2 - s_{i,j})/s_{i,j}^2}, \qquad \text{if } \textstyle\sum_{i=1}^{M}(q_{i,j}^2 - s_{i,j}) > 0, \tag{24}$$

$$\alpha_j = \infty, \qquad \text{otherwise.} \tag{25}$$

However, when $M > 1$, (23) may only correspond to one of many local maximum solutions of $\mathcal{L}(\alpha, \alpha_0)$. Therefore, we no longer perform an exact maximum likelihood estimation of α_j, but only increase $\mathcal{L}(\alpha, \alpha_0)$ at each iteration. This approximation allows much faster estimation of α_j than exactly solving (22), which requires solving of m polynomials of degree $2M - 1$ at each iteration.

The remaining formulas are similar to the those considered for the fast relevance vector machine (RVM), and therefore one may refer to [33] for more details. 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 adding terms to the model until all N nonzero weights have been added. Therefore, the complexity of the algorithm is more related to N than m. Further, by using the matrix inverse identity, the inverse operation in (12) has been implemented by iterative update formulae with reduced complexity (see the appendix of [33]). Detailed analysis shows that this fast algorithm has complexity $\mathcal{O}(mN^2)$, which is more efficient than the iterative solution mentioned in Sec. II-B1, which is $\mathcal{O}(tm^3)$, where t is the number of iterations, especially when the underlying signal is truly sparse ($N \ll m$).

III. Integrating Out Regression Noise Variance

To apply the fast algorithm discussed above, an initial guess of α_0 is required, and this value is then fixed thereafter to allow the iterative update formulae [33]. In this section we introduce a modified sparse-regression modeling for multi-task CS inversion. The algorithm integrates α_0 out, rather than seeking a point estimate of α_0, and the computation is solely concentrated on recovering the hyper-parameters α. This allows an efficient sequential optimization method that is applicable without the constraint of having a point estimate for α_0. While being more widely applicable, the modified fast algorithm is more robust to the parameter setting than the original RVM formulation in Sec. II. As we will see soon, this modified formalism and the fast inference algorithm can be derived in a manner parallel to [34].

We define a zero-mean Gaussian prior for each component of θ_i, and define a Gamma prior on the noise precision α_0:

$$p(\theta_i|\alpha,\alpha_0) = \prod_{j=1}^{M}\mathcal{N}(\theta_{i,j}\mid 0,\alpha_0^{-1}\alpha_j^{-1}), \tag{26}$$

$$p(\alpha_0|a,b) = Ga(\alpha_0|a,b). \tag{27}$$

Note that the differences between the formulation specified above and that in the original RVM [12] are: (i) α_0 is included in the prior of θ_i, and (ii) a Gamma prior of α_0 has been used. This modification allows the integration involved in the sequel to be performed analytically. Specifically, given α and CS measurements v_i, the likelihood function of θ_i may be expressed as

$$\begin{aligned} p(\theta_i|v_i,\alpha) &= \int p(\theta_i|v_i,\alpha,\alpha_0)p(\alpha_0|a,b)d\alpha_0 \\ &= \frac{\Gamma(a+m/2)\left[1+\frac{1}{2b}(\theta_i-\mu_i)^T\Sigma_i^{-1}(\theta_i-\mu_i)\right]^{-(a+m/2)}}{\Gamma(a)(2\pi b)^{m/2}|\Sigma_i|^{1/2}}, \end{aligned} \tag{28}$$

where

$$\mu_i = \Sigma_i \Phi_i^T v_i, \tag{29}$$

$$\Sigma_i = (\Phi_i^T \Phi_i + \mathrm{A})^{-1}, \tag{30}$$

with $\mathrm{A} = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_m)$. We note that, by integrating α_0 out, the likelihood function has been changed from a multivariate Gaussian distribution (10) to a multivariate Student-t distribution (28). Similarly, an empirical Bayesian approach (or type-II maximum likelihood) can be applied to estimate hyper-parameters α, i.e., seeking α to maximize the marginal likelihood, or equivalently, its logarithm:

$$\begin{aligned} \mathcal{L}(\alpha) &= \sum_{i=1}^{M} \log p(v_i|\boldsymbol{\alpha}) = \sum_{i=1}^{M} \log \int p(v_i|\theta_i, \alpha_0) p(\theta_i|\alpha, \alpha_0) p(\alpha_0|a, b) d\theta_i \, d\alpha_0 \\ &= -\frac{1}{2} \sum_{i=1}^{M} \left[(n_i + 2a) \log \left(v_i^T \mathrm{B}_i^{-1} v_i + b \right) + \frac{1}{2} \log |\mathrm{B}_i| \right] + \text{const}, \end{aligned} \tag{31}$$

with

$$\mathrm{B}_i = \mathrm{I} + \Phi_i \mathrm{A}^{-1} \Phi_i^T. \tag{32}$$

Before proceeding, we note that in addition to avoiding the need for a plug-in estimate of α_0, the marginalization of α_0 has the advantage of inducing a heavy-tailed distribution on the basis coefficients, as is apparent in (28). This allows for more robust shrinkage and borrowing of information, as some tasks can be outliers.

A. Iterative Optimization

Both direct differentiation and the EM algorithm can be applied to maximize (31) for a point estimate of $\boldsymbol{\alpha}$, yielding

$$\alpha_j = \frac{M}{\sum_{i=1}^{M} \mu_{i,j}^2 (n_i + 2a)/(v_i^T \mathrm{B}_i^{-1} v_i + 2b)}, \qquad j \in \{1, 2, \ldots, m\}. \tag{33}$$

This suggests an algorithm that iterates between (33) and (29)–(30) until convergence is achieved.

B. Fast Algorithm

The modified fast algorithm may be derived in a manner similar to the fast RVM algorithm [33]. Considering the dependence of $\mathcal{L}(\alpha)$ on a single hyperparameter α_j, $j \in \{1, 2, \ldots, M\}$, we may

decompose B_i in (32) as

$$\begin{aligned}\mathrm{B}_i &= \mathrm{I} + \Phi_i \mathrm{A}_i^{-1} \Phi_i^T = \mathrm{I} + \sum_{k \neq j} \alpha_k^{-1} \Phi_{i,k} \Phi_{i,k}^T + \alpha_j^{-1} \Phi_{i,j} \Phi_{i,j}^T \\ &= \mathrm{B}_{i,-j} + \alpha_j^{-1} \Phi_{i,j} \Phi_{i,j}^T, \end{aligned} \tag{34}$$

where $\mathrm{B}_{i,-j}$ is B_i with the contribution of basis function $\Phi_{i,j}$ removed. Matrix determinant and inverse identities may be used to express

$$|\mathrm{B}_i| = |\mathrm{B}_{i,-j}| \, |1 + \alpha_j^{-1} \Phi_{i,j}^T \mathrm{B}_{i,-j}^{-1} \Phi_{i,j}|, \tag{35}$$

$$\mathrm{B}_i^{-1} = \mathrm{B}_{i,-j}^{-1} - \frac{\mathrm{B}_{i,-j}^{-1} \Phi_{i,j} \Phi_{i,j}^T \mathrm{B}_{i,-j}^{-1}}{\alpha_j + \Phi_{i,j}^T \mathrm{B}_{i,-j}^{-1} \Phi_{i,j}}. \tag{36}$$

From this, we may write $\mathcal{L}(\alpha)$ as

$$\begin{aligned}\mathcal{L}(\alpha) &= -\frac{1}{2} \sum_{i=1}^{M} \left[(n_i + 2a) \log \left(\frac{1}{2} v_i^T \mathrm{B}_{i,-j}^{-1} v_i + b \right) + \log |\mathrm{B}_{i,-j}| \right] + \text{const} \\ &\quad -\frac{1}{2} \sum_{i=1}^{M} \left[\log(1 + \alpha_j^{-1} s_{i,j}) + (n_i + 2a) \log \left(1 - \frac{q_{i,j}^2 / g_{i,j}}{\alpha_j + s_{i,j}} \right) \right] \\ &= \mathcal{L}(\alpha_{-j}) + \ell(\alpha_j), \end{aligned} \tag{37}$$

where $\alpha_{i,-j}$ is the same as α_i except the jth component is removed, and we have defined

$$s_{i,j} \triangleq \Phi_{i,j}^T \mathrm{B}_{i,-j}^{-1} \Phi_{i,j}, \quad q_{i,j} \triangleq \Phi_{i,j}^T \mathrm{B}_{i,-j}^{-1} v_i, \quad \text{and} \quad g_{i,j} \triangleq v_i^T \mathrm{B}_{i,-j}^{-1} v_i + 2b. \tag{38}$$

Differentiating $\ell(\alpha_j)$ with respect to α_j and setting the result to zero yields the following after algebra:

$$\sum_{i=1}^{M} \frac{(n_i + 2a) q_{i,j}^2 / g_{i,j} - s_{i,j} - s_{i,j}(s_{i,j} - q_{i,j}^2 / g_{i,j}) / \alpha_j}{(\alpha_j + s_{i,j})(\alpha_j + s_{i,j} - q_{i,j}^2 / g_{i,j})} = 0. \tag{39}$$

Again, except for the trivial solution $\alpha_j = \infty$, the other solutions of (39) cannot be expressed analytically. We thus assume that $\alpha_j \ll s_{i,j}$ (this has generally been found to be true numerically or causes more iterations upon convergence) and the denominator of (39) is now relative invariant with respect to α_j. Therefore, we may approximate another solution as

$$\alpha_j \approx \frac{M}{\sum_{i=1}^{M} \frac{(n_i + 2a) q_{i,j}^2 / g_{i,j} - s_{i,j}}{s_{i,j}(s_{i,j} - q_{i,j}^2 / g_{i,j})}}. \tag{40}$$

Evidently, when $M = 1$, (40) recovers the exact solution of (39). Similar analysis of $\ell(\alpha_j)$ as in [34] when $M = 1$ shows that $\mathcal{L}(\alpha)$ has a unique maximum with respect to α_j:

$$\alpha_j \approx \frac{M}{\sum_{i=1}^{M} \frac{(n_i+2a)q_{i,j}^2/g_{i,j}-s_{i,j}}{s_{i,j}(s_{i,j}-q_{i,j}^2/g_{i,j})}}, \qquad \text{if } \sum_{i=1}^{M} \frac{(n_i+2a)q_{i,j}^2/g_{i,j}-s_{i,j}}{s_{i,j}(s_{i,j}-q_{i,j}^2/g_{i,j})} > 0, \tag{41}$$

$$\alpha_j = \infty, \qquad \text{otherwise.} \tag{42}$$

As before, when $M > 1$, (40) may only correspond to one of many local maximum solutions of $\mathcal{L}(\alpha)$. Therefore, we no longer perform an exact maximum likelihood estimate of α_j, but only increase $\mathcal{L}(\alpha)$ at each iteration.

Recall that setting $\alpha_j = \infty$ is equivalent to $\theta_{i,j} = 0$, and hence removing $\Phi_{i,j}$ from the representation; hence, (41)–(42) controls the addition and deletion of particular $\Phi_{i,j}$ from the signal representation. If we perform these operations sequentially for varying j, we realize an efficient learning algorithm.

In practice, it is relatively straightforward to compute $s_{i,j}$ and $q_{i,j}$ for all the basis vector $\Phi_{i,j}$, including those not currently utilized in the model (i.e., for which $\alpha_j = \infty$). These quantities can be computed by maintaining and updating values of

$$S_{i,j} = \Phi_{i,j}^T \mathrm{B}_i^{-1} \Phi_{i,j}, \quad Q_{i,j} = \Phi_{i,j}^T \mathrm{B}_i^{-1} v_i, \quad \text{and} \quad G_i = v_i^T \mathrm{B}_i^{-1} v_i + 2b, \tag{43}$$

and from these it follows simply:

$$s_{i,j} = \frac{\alpha_j S_{i,j}}{\alpha_j - S_{i,j}}, \quad q_{i,j} = \frac{\alpha_j Q_{i,j}}{\alpha_j - S_{i,j}}, \quad \text{and} \quad g_{i,j} = G_i + \frac{Q_{i,j}^2}{\alpha_j - S_{i,j}}. \tag{44}$$

Note that when $\alpha_j = \infty$, $s_{i,j} = S_{i,j}$, $q_{i,j} = Q_{i,j}$ and $g_{i,j} = G_i$. Further, it is convenient to utilize the Woodbury identity to obtain the quantities of interest:

$$S_{i,j} = \Phi_{i,j}^T \Phi_{i,j} - \Phi_{i,j}^T \Phi_i \Sigma_i \Phi_i^T \Phi_{i,j}, \tag{45}$$

$$Q_{i,j} = \Phi_{i,j}^T v_i - \Phi_{i,j}^T \Phi_i \Sigma_i \Phi_i^T v_i, \tag{46}$$

$$G_i = v_i^T v_i - v_i^T \Phi_i \Sigma_i \Phi_i^T v_i + 2b. \tag{47}$$

Here quantities Φ_i and Σ_i contain only those basis vectors that are currently included in the model, and the computation thus scales as the cube of that measure, which is typically only a very small fraction of m. Furthermore, these quantities can also be calculated via the update formulae, as shown in the Appendix, with reduced computation. Note that similar update formulae are applied to the original fast RVM algorithm [33] when α_0 is fixed. However, our modified fast algorithm is applicable without this

constraint.

IV. Example Results

We denote the fast algorithm in Sec. II-B as BCS, and the fast algorithm in Sec. III-B as BCS*, and test the performance of BCS and BCS* on both single-task (ST) and multi-task (MT) CS inverse problems. To be concise, in the example CS-reconstruction figures that follow we only present the BCS* results , and the full quantitative performance comparison between BCS and BCS* is summarized in tables. For a fair comparison between BCS and BCS*, we initialize $\alpha_0 = 10^2/\mathrm{std}(v)^2$ and fix this value thereafter (for the fast algorithm) for BCS; with regard to BCS*, we set $a = 10^2/\mathrm{std}(v)^2$ and $b = 1$ such that the mean of the Gamma prior $p(\alpha_0|a,b)$[1] is aligned with the fixed value of α_0 in BCS. In the experiments we evaluate the reconstruction error as $\|\hat{u}_{\mathrm{method}} - u\|_2/\|u\|_2$. The Matlab code is available online at `http://www.ece.duke.edu/~shji/BCS.html`. All the computations presented here were performed on a 3.4GHz Pentium machine.

A. 1D Signals

In the first example we consider $M = 2$ signals of length $m = 512$, each containing 20 spikes created by choosing 20 locations at random and then putting ± 1 at these points (Figs. 2(a-b)). The two original signals are created such that they have 75% spikes at the same positions, but all have random ± 1 amplitudes. The projection matrix Φ_i is constructed by first creating a $n_i \times m$ matrix with i.i.d. draws of a Gaussian distribution $\mathcal{N}(0,1)$, and then the rows of Φ_i are normalized to unit amplitude. Zero-mean Gaussian noise with standard deviation $\sigma_0 = 0.005$ is added to each of the n_i measurements that define the data v_i. In the experiment $n_1 = 90$, $n_2 = 70$ and the reconstructions are implemented by ST-CS and MT-CS, respectively.

Figures 2(c-d) demonstrate the reconstruction results with BCS* for single-task inference. Because of lack of enough measurements (n_i is smaller than a minimum quantity required for faithful reconstruction [5], [6]), the reconstructed signals are highly noisy. However, since two original signals are not statistically independent, multi-task CS is able to take advantage of the inter-relationships and yields almost perfect reconstructions (Figs. 2(e-f)). The results of BCS are very similar to BCS*, and therefore are omitted here.

To study how the similarity between the original signals affects the reconstruction performance of MT-CS, in the second experiment, we use the same dataset as in Fig. 2 and study the performance

[1]The variance of the Gamma prior is a/b^2.

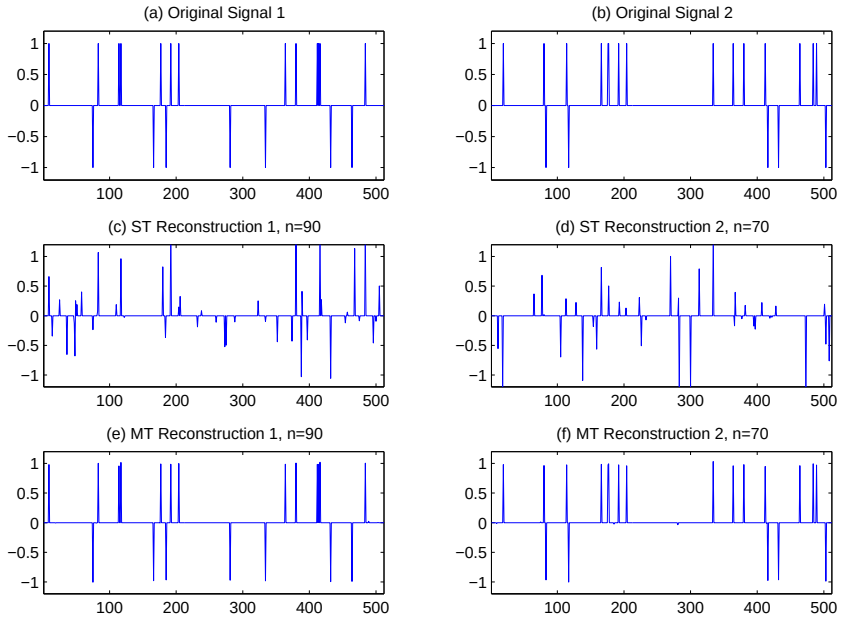

Fig. 2. Reconstruction of the *Spikes* of length $m = 512$. The two original signals have 75% spikes at the same positions, but all have random ± 1 amplitudes. (a-b) Original signals; (c-d) reconstructed signals by ST-BCS*; (e-f) reconstructed signals by MT-BCS*.

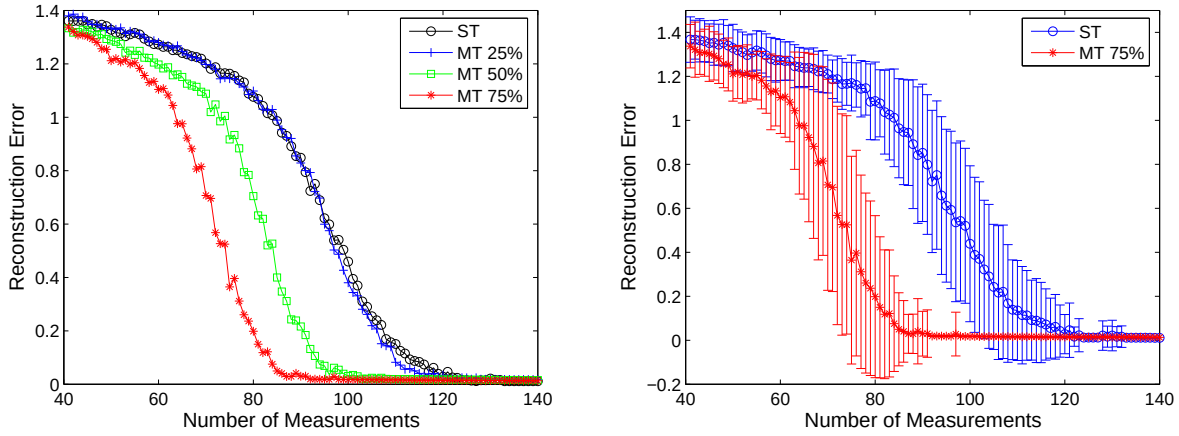

Fig. 3. Reconstruction errors of ST-BCS* and MT-BCS* as a function of increasing n. The two original signals have 75%, 50%, or 25% spikes at the same positions, with random ± 1 amplitudes. The results are averaged over 100 runs. (a) The average reconstruction errors for 75%, 50% and 25% similarity; (b) the variance of reconstruction errors for 75% similarity.

of BCS* for different similarity levels, i.e., 75%, 50% or 25% spikes are at the same locations. For each similarity level, starting at the 41st measurement, after each random measurement is conducted, the associated reconstruction errors are computed by ST-BCS* and MT-BCS*, until a total 140 measurements are conducted for each of the two original signals. Because of the randomness in the experiment (i.e., the random CS measurements and the random locations and ± 1 amplitudes of the spikes), 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 MT-BCS* is much smaller than that of the ST-BCS*, when the similarities are at 75% and 50%. However, when the similarity is 25%, the improvements are minor or none. This suggests that for multi-task CS to be superior, the original signals should have at least some level of similarity. In this experiment, the BCS results are very similar to BCS*, therefore, are omitted here. It is also important to note that the multi-task CS procedure mimics single-task inference when the signals are not sufficiently close, such that one doesn't undermine final performance if multi-task CS is attempted but not appropriate.

B. 2D Images

In the following set of experiments, the performance of MT-CS is compared to ST-CS on three example problems. All the projection matrices Φ considered here are drawn from a uniform spherical distribution [35].

1) Random-Bars: Figure 4 shows the reconstruction results for *Random-Bars*, where Fig. 4(a) is from [35] and the other two images (b-c) are modified from (a) to represent similar tasks for simultaneous CS inversion, i.e., the intensities of all the rectangles in (b-c) are randomly permuted from (a), and the positions of all the rectangles are shifted by distances randomly sampled from a uniform distribution in $[-10, 10]$. All three original images have the size 1024×1024. We used the Haar wavelet expansion, which is well suited to images of this type, with a coarsest scale $j_0 = 3$, and a finest scale $j_1 = 6$. Figures 4(a-c) shows the result of linear reconstruction with $n = 4096$ samples, which represents the best performance that could be achieved by all the CS implementations used. Figures 4(d-f) have results of ST-BCS* by using the hybrid CS scheme [35] with $n = 670$ compressed samples for each task, whereas Figs. 4(g-i) have the results of MT-BCS*. The performance comparison between BCS and BCS* is summarized in I.

It is demonstrated that MT-CS yields a better reconstruction performance than that of ST-CS, both for BCS and BCS*; comparing the reconstruction errors of BCS and BCS* for the case of single task learning, ST-BCS* is markedly better than ST-BCS, whereas for multi-task learning MT-BCS* is only slightly better than MT-BCS; this is likely because in multi-task learning one utilizes more data, and therefore the differences manifested by an improved algorithm are less apparent. These results indicate that, in general, the integrating out of α_0, as in BCS*, may be a preferred approach rather than estimating α_0 as in BCS, particularly when the available data are not abundant.

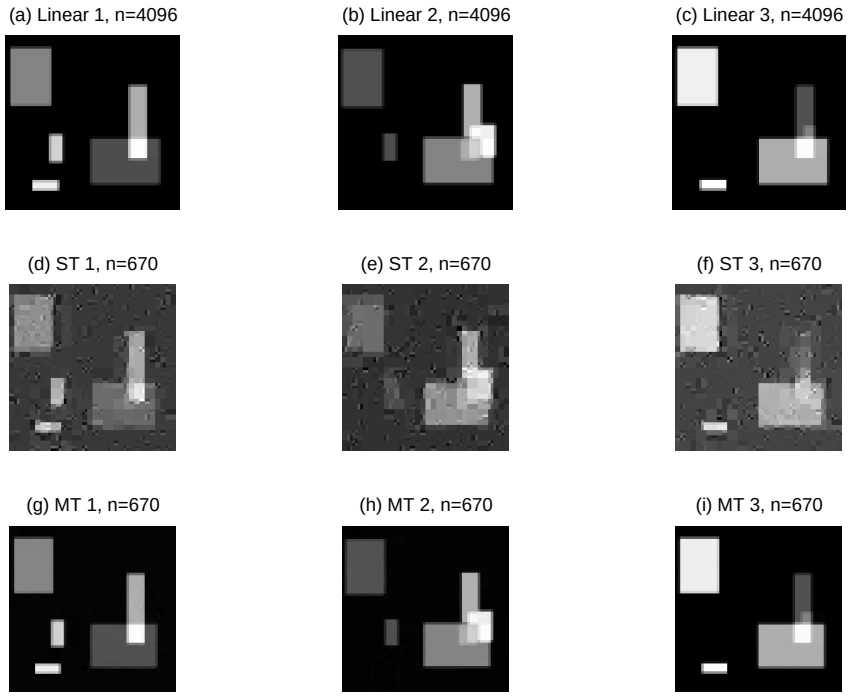

Fig. 4. Reconstruction of *Random-Bars* with hybrid CS. (a-c) Linear reconstructions of three original images. Example (a) is from [35], and (b-c) are the modified images from (a) by us to represent similar tasks for simultaneous CS inversion. The intensities of all the rectangles in (b-c) are randomly permuted from (a), and the positions of all the rectangles are shifted by distances randomly sampled from a uniform distribution in $[-10, 10]$. (d-f) reconstructed images by ST-BCS*; (g-i) reconstructed images by MT-BCS*.

TABLE I
RECONSTRUCTION PERFORMANCES OF LINEAR, ST-CS AND MT-CS ON *Random-Bars*.

	Recon. Error (%)			Run Time (secs)		
	Task 1	Task 2	Task 3	Task 1	Task 2	Task 3
ST-BCS	0.4439	0.3619	0.3674	63.91	40.72	24.53
MT-BCS	0.2319	0.2281	0.1977	67.76 per task		
ST-BCS*	0.3678	0.3502	0.3038	57.91	39.99	33.33
MT-BCS*	0.2277	0.2181	0.1936	39.74 per task		
Linear	0.2271	0.2178	0.1936	0		

2) MRI Images: Figure 5 shows the reconstruction results for *MRI Images*, which includes five image slices of a human head. All five original images have the size 128×128. We used a hybrid CS scheme [35] for image reconstruction, with a coarsest scale $j_0 = 3$, and a finest scale $j_1 = 6$ on the "Daubechies8" wavelet. Figure 5(a-e) show the results of linear reconstruction with $n = 4096$ samples, which represents the best performance that could be achieved by all the CS implementations used. Figures 5(f-j) have results for the ST-BCS* with $n = 1636$ compressed samples for each task, whereas Figs. 5(k-o) have

the results for MT-BCS*. The full performance comparison between BCS and BCS* are summarized in table II. The relative performance of ST-BCS to MT-BCS, and between BCS and BCS*, are consistent with the above *Random-Bars* results.

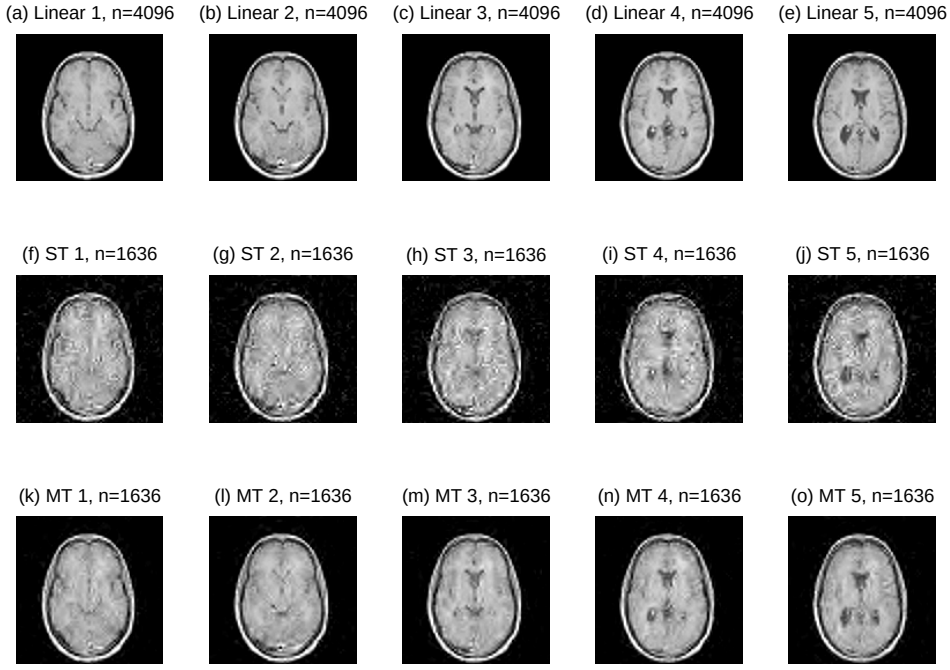

Fig. 5. Reconstruction of MRI images with hybrid CS. (a-e) Linear reconstructions of five original MRI images that are image slices of a human head; (f-j) reconstructed images by ST-BCS*; (k-o) reconstructed images by MT-BCS*.

TABLE II
RECONSTRUCTION PERFORMANCES OF LINEAR, ST-CS AND MT-CS ON MRI IMAGES.

	Recon. Error (%)					Run Time (secs)				
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 1	Task 2	Task 3	Task 4	Task 5
ST-BCS	0.2838	0.2859	0.2808	0.2943	0.2890	263.41	145.35	150.88	256.01	78.26
MT-BCS	0.2019	0.2019	0.2081	0.2079	0.2191	405.77 per task				
ST-BCS*	0.2515	0.2531	0.2658	0.2645	0.2744	262.70	387.95	821.45	158.80	498.38
MT-BCS*	0.1937	0.1937	0.1998	0.1999	0.2099	332.63 per task				
Linear	0.1690	0.1692	0.1777	0.1777	0.1851	0				

3) Still Images from Video Sequence: Figure 6 shows the reconstruction results for *Duke Video Images*, which are five snapshots from a web-camera. All five original images have the size 240×256. We used a hybrid CS scheme [35] for image reconstruction, with a coarsest scale $j_0 = 3$, and a finest scale $j_1 = 6$ on the "Daubechies8" wavelet. Figure 5(a-e) show the results of linear reconstruction with $n = 4096$

samples, which represents the best performance that could be achieved by all the CS implementations used. Figures 5(f-j) have results for the ST-BCS* with $n = 1717$ compressed samples for each task, whereas Figs. 5(k-o) have the results for MT-BCS*. The full performance comparison between BCS and BCS* is summarized in table III. Again, the conclusions on the relative performance of the different algorithms are consistent with those from the examples above.

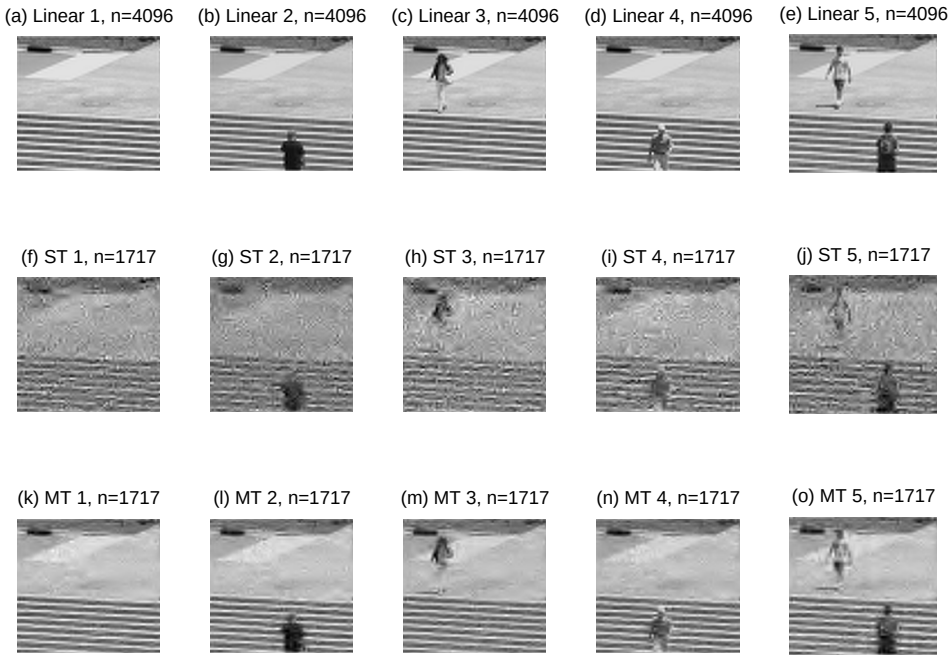

Fig. 6. Reconstruction of video images with hybrid CS. (a-e) Linear reconstructions of five image snapshots from a web-camera; (f-j) reconstructed images by ST-BCS*; (k-o) reconstructed images by MT-BCS*.

TABLE III
RECONSTRUCTION PERFORMANCES OF LINEAR, ST-CS AND MT-CS ON VIDEO IMAGES.

	Recon. Error (%)					Run Time (secs)				
	Task 1	Task 2	Task 3	Task 4	Task 5	Task 1	Task 2	Task 3	Task 4	Task 5
ST-BCS	0.2217	0.2090	0.2317	0.2059	0.2221	380.69	776.94	351.91	948.72	319.54
MT-BCS	0.1595	0.1531	0.1605	0.1487	0.1643	430.30 per task				
ST-BCS*	0.2029	0.1993	0.2146	0.1867	0.2080	353.11	328.68	306.40	227.92	529.63
MT-BCS*	0.1591	0.1524	0.1601	0.1485	0.1630	240.63 per task				
Linear	0.1539	0.1449	0.1518	0.1407	0.1508	0				

V. CONCLUSIONS

This paper has examined the application of multi-task learning to compressive sensing. This problem has been analyzed within the context of a hierarchical Bayesian framework, via sparse regression. This setting allows the sharing of information collected in multiple compressive measurements, to enhance the CS reconstructions. The framework developed here is particularly relevant for problems in which the images under test have a relatively high degree of similarity, e.g., when performing CS inversion of multiple medical images of the same body part, with the multiple CS measurements taken from the same or different individuals. Further, this setting is well suited to video data, where consecutive images are expected to have a high degree of statistical similarity.

In addition to introducing multi-task CS, a modified fast inference algorithm has been introduced. Specifically, a method has been introduced whereby the noise variance in the regression analysis is integrated out analytically. In previous sparse regression analyses of the type considered here a point estimate for the noise variance has been performed, in a ML/MAP sense, along with a ML/MAP estimate of the hyper-parameters of the sparseness-promoting prior. By integrating out the noise variance analytically, the associated uncertainty in this parameter is retained throughout, and one less parameter need be estimated. This modified fast algorithm has been compared with the original fast RVM algorithm [33], and it has been demonstrated to improve performance, both in single-task and multi-task CS. In fact, the advantages of this new algorithm were shown to be more pronounced in single-task CS (where CS inversion of each image is performed independently), with this attributed to the fact that less data are utilized in such cases; when appropriate, the sharing of data inherent to multi-task learning, between the different tasks, reduces the amount of data required for any one task, and is likely to improve the accuracy of an ML/MAP estimate.

A significant limitation of the multi-task CS analysis as considered here is the sensitivity of the wavelet coefficients to shifts in the image. This limitation is manifested because the sharing mechanism, as implemented, is directed toward the prior on the wavelet coefficients. Consider two images, with the same basic object (e.g., picture of person) in both images, but the object is significantly shifted in one image with respect to the other. While a human viewing these two images would be able to share information by looking at both, the multi-task CS algorithm presented here would not share information, once the object shift between the two images is sufficient. This suggests that, to generalize the multi-task CS, the sharing mechanism should not be directly on the wavelet coefficients, but rather imposed at a higher level. This is an area of open research, but one may conjecture about possible future directions.

For example, rather than placing the shared prior on the wavelet coefficients, one may share a prior on the *statistics* of quadtrees [2]. The sharing in this case is imposed not at the wavelet-coefficient level, but at the quadtree level. Considering the previous example again, the same object shifted within an image may have similar local quadtree statistics, although the location of the similar quadtrees are shifted within the image, commensurate with the associated object shift in the original image. Statistical models such as the hidden Markov tree [36] may be used to model the statistics of the quadtrees, and the multi-task sharing mechanisms may be implemented using more-sophisticated MTL tools than those investigated here. For example, the Dirichlet process [37] has proven to be a very effective tool for multi-task learning; this type of model is also within the hierarchical Bayesian family, but with far more sophistication and generality than that considered here. Future research may be considered to extend these techniques to multi-task compressive sensing with emphasis on computational efficiency and sparse solutions.

ACKNOWLEDGEMENT

The authors thank I. Pruteanu for providing the video images. This work was supported by the Office of Naval Research.

APPENDIX

In the implementation of the fast algorithm in Sec. III-B, it is necessary to recompute Σ_i, μ_i, and all quantities $s_{i,j}$, $q_{i,j}$ and $g_{i,j}$. For the sequential nature of the fast algorithm, these quantities can be calculated iteratively. In addition, we must calculate the increase or decrease of the marginal likelihood $\mathcal{L}(\alpha) = \sum_{i=1}^{M} \mathcal{L}_i(\alpha)$ according to which basis functions are added, deleted or re-estimated. Efficient calculations of these quantities are given below.

A. Notation

The fast algorithm operates in a constructive manner, i.e., at each step t it may add a basis to the model, or delete a basis from the model, or re-estimate the parameters of the model. Therefore, Φ_i as used below need only comprise columns of included basis functions. Denote the number of the basis functions in Φ_i at step t as m_t, so Φ is of size $n_i \times m_t$. Similarly, Σ_i and μ_i are computed only for the "current" basis and therefore are of order m_t (all other entries in the "full" version of Σ_i and μ_i would be zero). The integer $j \in \{1, 2, \ldots, m\}$ is used to index the single basis function for which α_j is to be updated, and the integer $k \in \{1, 2, \ldots, m_t\}$ to denote the index within the current basis that corresponds to j. The index $l \in \{1, 2, \ldots, m\}$ ranges over all basis functions, including those not currently utilized in the model. For convenience, define $K_i = n_i + 2a$. Updated quantities are denoted by a tilde (e.g., $\widetilde{\alpha}_i$).

B. Adding a new basis function

$$2\Delta\mathcal{L}_i = \log\frac{\alpha_j}{\alpha_j+s_{i,j}} - K_i\log\left(1-\frac{q_{i,j}^2/g_{i,j}}{\alpha_j+s_{i,j}}\right), \tag{48}$$

$$\widetilde{\Sigma}_i = \begin{bmatrix} \Sigma_i + \Sigma_{i,(jj)}\Sigma_i\Phi_i^T\Phi_{i,j}\Phi_{i,j}^T\Phi_i\Sigma_i & -\Sigma_{i,(jj)}\Sigma_i\Phi_i^T\Phi_{i,j} \\ -\Sigma_{i,(jj)}(\Sigma_i\Phi_i^T\Phi_{i,j})^T & \Sigma_{i,(jj)} \end{bmatrix}, \tag{49}$$

$$\widetilde{\mu}_i = \begin{bmatrix} \mu_i - \mu_{i,j}\Sigma_i\Phi_i^T\Phi_{i,j} \\ \mu_{i,j} \end{bmatrix}, \tag{50}$$

$$\widetilde{S}_{i,l} = S_{i,l} - \Sigma_{i,(jj)}(\Phi_{i,l}^T e_{i,j})^2, \tag{51}$$

$$\widetilde{Q}_{i,l} = Q_{i,l} - \mu_{i,j}(\Phi_{i,l}^T e_{i,j}), \tag{52}$$

$$\widetilde{G}_i = G_i - \Sigma_{i,(jj)}(v_i^T e_{i,j})^2. \tag{53}$$

where $\Sigma_{i,(jj)} = (\alpha_{i,j}+S_{i,j})^{-1}$, $\mu_{i,j} = \Sigma_{i,(jj)}Q_{i,j}$ and we define $e_{i,j} \triangleq \Phi_{i,j} - \Phi_i\Sigma_i\Phi_i^T\Phi_{i,j}$.

C. Re-estimating a basis function

Defining $\gamma_{i,k} \triangleq (\Sigma_{i,(kk)} + (\widetilde{\alpha}_j - \alpha_j)^{-1})^{-1}$ and $\Sigma_{i,k}$ as the k-th column of Σ_i:

$$2\Delta\mathcal{L}_i = (K_i-1)\log(1+S_{i,j}(\widetilde{\alpha}_j^{-1}-\alpha_j^{-1})) + K_i\log\frac{[(\alpha_j+s_{i,j})g_{i,j}-q_{i,j}^2]\widetilde{\alpha}_j}{[(\widetilde{\alpha}_j+s_{i,j})g_{i,j}-q_{i,j}^2]\alpha_j}, \tag{54}$$

$$\widetilde{\Sigma}_i = \Sigma_i - \gamma_{i,k}\Sigma_{i,k}\Sigma_{i,k}^T, \tag{55}$$

$$\widetilde{\mu}_i = \mu_i - \gamma_{i,k}\mu_{i,k}\Sigma_{i,k}, \tag{56}$$

$$\widetilde{S}_{i,l} = S_{i,l} + \gamma_{i,k}(\Sigma_{i,k}^T\Phi_i^T\Phi_{i,l})^2, \tag{57}$$

$$\widetilde{Q}_{i,l} = Q_{i,l} + \gamma_{i,k}\mu_{i,k}(\Sigma_{i,k}^T\Phi_i^T\Phi_{i,l}), \tag{58}$$

$$\widetilde{G}_i = G_i + \gamma_{i,k}(\Sigma_{i,k}^T\Phi_i^T v_i)^2. \tag{59}$$

D. Deleting a basis function

$$2\Delta\mathcal{L}_i = -K_i \log\left(1 + \frac{Q_{i,j}^2/G_i}{\alpha_j - S_{i,j}}\right) - \log\left(1 - \frac{S_{i,j}}{\alpha_j}\right), \tag{60}$$

$$\widetilde{\Sigma}_i = \Sigma_i - \frac{1}{\Sigma_{i,(kk)}}\Sigma_{i,k}\Sigma_{i,k}^T, \tag{61}$$

$$\widetilde{\mu}_i = \mu_i - \frac{\mu_{i,k}}{\Sigma_{i,(kk)}}\Sigma_{i,k}, \tag{62}$$

$$\widetilde{S}_{i,l} = S_{i,l} + \frac{1}{\Sigma_{i,(kk)}}(\Sigma_{i,k}^T\Phi_i^T\Phi_{i,l})^2, \tag{63}$$

$$\widetilde{Q}_{i,l} = Q_{i,l} + \frac{\mu_{i,k}}{\Sigma_{i,(kk)}}(\Sigma_{i,k}^T\Phi_i^T\Phi_{i,l}), \tag{64}$$

$$\widetilde{G}_i = G_i + \frac{1}{\Sigma_{i,(kk)}}(\Sigma_{i,k}^T\Phi_i^T v_i)^2. \tag{65}$$

Following updates (61) and (62), the appropriate row and/or column k is removed from $\widetilde{\Sigma}_i$ and $\widetilde{\mu}_i$.

REFERENCES

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

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

[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] 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.

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

[9] 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.

[10] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," Jan. 2007, Preprint.

[11] M. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," 2007, Preprint.

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

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

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

[17] J. Baxter, "Learning internal representations," in *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1995.

[18] ——, "A model of inductive bias learning," *Journal of Artificial Intelligence Research*, 2000.

[19] K. Yu, A. Schwaighofer, V. Tresp, W.-Y. Ma, and H. Zhang, "Collaborative ensemble learning: Combining collaborative and content-based information filtering via hierarchical Bayes," in *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, 2003.

[20] K. Yu, V. Tresp, and S. Yu, "A nonparametric hierarchical Bayesian framework for information filtering," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

[21] K. Yu, V. Tresp, and A. Schwaighofer, "Learning Gaussian processes from multiple tasks," in *Proc. of the 22nd International Conference on Machine Learning (ICML 22)*, 2005.

[22] J. Zhang, Z. Ghahramani, and Y. Yang, "Learning multiple related tasks using latent independent component analysis," in *Advances in Neural Information Processing Systems 18*, 2005.

[23] N. D. Lawrence and J. C. Platt, "Learning to learn with the informative vector machine," in *Proc. of the 21st International Conference on Machine Learning (ICML 21)*, 2004.

[24] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, pp. 1817–1853, 2005.

[25] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[26] G. V. Glass, "Primary, secondary and meta-analysis of research," *Educatinal Researcher*, vol. 5, 1976.

[27] D. Burr and H. Doss., "A Bayesian semiparametric model for random-effects meta-analysis," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 242–251, Mar. 2005.

[28] F. Dominici, G. Parmigiani, R. Wolpert, and K. Reckhow, "Combining information from related regressions," *Journal of Agricultural, Biological, and Environmental Statistics*, vol. 2, no. 3, pp. 294–312, 1997.

[29] P. D. Hoff, "Nonparametric modeling of hierarchically exchangeable data," University of Washington Statistics Department, Tech. Rep. 421, 2003.

[30] P. Müller, F. Quintana, and G. Rosner, "A method for combining inference across related nonparametric Bayesian models," *Journal of the Royal Statistical Society Series B*, vol. 66, no. 3, pp. 735–749, 2004.

[31] B. K. Mallick and S. G. Walker, "Combining information from several experiments with nonparametric priors," *Biometrika*, vol. 84, no. 3, pp. 697–706, 1997.

[32] B. P. Carlin and T. A. Louis, *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall-CRC, 2000.

[33] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," in *Proc. of the 9th International Workshop on AIStats*, C. M. Bishop and B. J. Frey, Eds., 2003.

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

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

[36] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, April 1998.

[37] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.