

Bayesian Multi-Task Compressive Sensing with Dirichlet Process Priors

¹Yuting Qi, ¹Dehong Liu, ²David Dunson and ¹Lawrence Carin

¹Department of Electrical and Computer Engineering

² Department of Statistical Science

Duke University, Durham, NC, 27708

Email: {yuting, liudh, lcarin}@ee.duke.edu, dunson@stat.duke.edu

Abstract

Compressive sensing (CS) is an emerging field that, under appropriate conditions, can significantly reduce the number of measurements required for a given signal. Specifically, if the m -dimensional signal \mathbf{u} is sparse in an orthonormal basis represented by the $m \times m$ matrix Ψ , then one may infer \mathbf{u} based on $n \ll m$ projection measurements. If $\mathbf{u} = \Psi\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ are the sparse coefficients in basis Ψ , then the CS measurements are represented by $\mathbf{v} = \Phi\boldsymbol{\theta}$, where \mathbf{v} is an n -dimensional vector and Φ is an $n \times m$ projection matrix. There are several ways in which the matrix Φ may be constituted, and one typically inverts for the signal \mathbf{u} by solving $\mathbf{v} = \Phi\boldsymbol{\theta}$ under the constraint that $\boldsymbol{\theta}$ is sparse (with this often performed with l_1 regularization). In many applications, one is interested in multiple signals $\{\mathbf{u}_i\}_{i=1,M}$ that may be measured in multiple CS-type measurements, where here each \mathbf{u}_i corresponds to a sensing “task”. It is possible to improve the CS performance (*e.g.*, requiring fewer total CS measurements) by exploiting the statistical inter-relationships of the associated $\{\mathbf{v}_i\}_{i=1,M}$ CS measurements, *jointly* inverting for the M underlying signals. In this paper we propose a novel multi-task compressive sensing framework based on a Bayesian formalism, where a sparseness prior is adopted. The key challenge is that not all of the measured $\{\mathbf{v}_i\}_{i=1,M}$ are necessarily appropriate for sharing when performing inversion, and one must therefore infer what sharing of data across the M “tasks” is appropriate. Toward this end, a Dirichlet process (DP) prior is employed, which provides a principled means of inferring the appropriate sharing mechanisms (*i.e.*, it infers how best to cluster the M CS measurements, with CS inversion effectively performed separately within each cluster). The posteriors of the sparse signals as well as the sharing mechanism are inferred among all CS tasks. A variational Bayesian (VB) inference algorithm is employed to estimate the full posterior on the model parameters, and an even more efficient simplified VB DP algorithm is also considered.

Index Terms

Compressive sensing (CS), Multi-task learning, Dirichlet Process Priors, Sparse Bayesian learning, Variational Bayes

I. INTRODUCTION

Since the late 1940s and the pioneering work of Shannon [31], research in data compression has achieved significant progress, especially over the last two decades in which researchers have considered sparse signal representations in terms of orthonormal basis functions (*e.g.*, the wavelet transform [10]). For example, consider an m -dimensional real-valued signal \mathbf{u} and assume an $m \times m$ orthonormal basis matrix Ψ ; we may then express $\mathbf{u} = \Psi\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is an m -dimensional

column vector of weighting coefficients. For most natural signals there exists an orthonormal basis Ψ such that θ is sparse (*e.g.*, JPEG [37] is based on a DCT basis, and JPEG2000 [8] on a wavelet basis). Consider now an approximation to \mathbf{u} , $\hat{\mathbf{u}} = \Psi\hat{\theta}$, where $\hat{\theta}$ approximates θ by retaining the largest N coefficients and setting the remaining $m - N$ coefficients to zero; due to the aforementioned sparseness properties, $\|\mathbf{u} - \hat{\mathbf{u}}\|^2$ is typically very small even for $N \ll m$. This property has led to the development of state-of-art transform-coding compression techniques [27][29].

Conventional techniques such as JPEG and JPEG2000 require one to first measure the m -dimensional signal \mathbf{u} , then compute the complete set of transform coefficients, followed (effectively) by locating the largest N coefficients and encoding them, and finally discarding all other coefficients [8]. This sample-then-compress framework is often wasteful since the signal acquisition is potentially expensive, and only a small amount of data N is eventually required for the accurate approximation $\hat{\mathbf{u}}$. For example, measurements in some applications (*e.g.*, medical scanners and radars) are expensive from the standpoint of energy or time [1]. One may therefore consider the following fundamental question: Since the signal \mathbf{u} is typically compressible, is it possible to directly measure the informative part of the signal? Recent research in the field of compressive sensing shows that this is indeed possible [6][12][36].

Exploiting the same sparseness properties of \mathbf{u} employed in transform coding ($\mathbf{u} = \Phi\theta$ with θ sparse), in compressive sensing one measures $\mathbf{v} = \Phi\theta$, where \mathbf{v} is an n -dimensional vector with $n < m$, and Φ is the $n \times m$ sensing matrix. There are several ways in which Φ may be constituted, with the reader referred to [12] for details. In most cases Φ is represented as $\Phi = \mathbf{T}\Psi$, where \mathbf{T} is an $n \times m$ matrix with components constituted randomly [36]; hence, the CS measurements correspond to projections of \mathbf{u} with the rows of \mathbf{T} : $\mathbf{v} = \mathbf{T}\mathbf{u} = \mathbf{T}\Psi\theta = \Phi\theta$. To recover θ (and hence \mathbf{u}) one must solve the under-determined problem $\mathbf{v} = \Phi\theta$, while exploiting the aforementioned assumption that θ is sparse. Assuming the signal \mathbf{u} is N -sparse in Ψ , implying that the coefficients θ only have N nonzero values [6] [12], Candès, Romberg and Tao in [7] show that, with overwhelming probability, θ (and hence \mathbf{u}) is recovered via

$$\min \|\theta\|_{l_1}, \quad \text{s.t.}, \quad \mathbf{v} = \Phi\theta, \quad (1)$$

if the number of CS measurements $n > C \cdot N \cdot \log m$ (C is a small constant); if N is small (*i.e.*, if \mathbf{u} is highly compressible in the basis Ψ) then $n \ll m$. In practice the signal \mathbf{u} is not exactly

sparse, but a large number of coefficients in the basis Ψ may be discarded with minimal error in reconstructing \mathbf{u} ; in this practical case the CS framework has also been shown to operate effectively.

The problem in (1) may be solved by linear programming [28] and greedy algorithms [35] [13]. A Bayesian compressive sensing (BCS) methodology is proposed in [24], by posing the CS inversion problem as a linear-regression problem with a sparseness prior on the regression weights θ . One of the advantages of BCS is that a full posterior density function is provided for θ , yielding a measure of uncertainty in the approximation to \mathbf{u} (in practice if one attempts to approximate \mathbf{u} , it could be better to adaptively terminate CS measurements based on error bars on $\Psi\theta$). Furthermore, the BCS framework may be extended to jointly invert multiple CS measurements $\{\mathbf{v}_i\}_{i=1,M}$, with associated underlying sparse weights $\{\theta_i\}_{i=1,M}$ [23]. Each CS measurement $\mathbf{v}_i = \Phi_i\theta_i$ represents a sensing “task”, and the objective is to jointly invert for all $\{\theta_i\}_{i=1,M}$, through an appropriate sharing of information between the M data collections. Assuming we have M CS tasks, one may potentially reduce the number of measurements required for each task by exploiting the statistical relationships among the tasks. For example, “Distributed Compressed Sensing” (DCS) seeks to jointly recover the nonzero wavelet coefficients [2] across multiple CS measurements. Alternatively, in [39] Wipf and Rao considered an empirical Bayesian strategy “Simultaneous Sparse Approximation”. Of most relevance to the current paper, a hierarchical Bayesian model has been designed for multi-task CS [23], in which all tasks are assumed to have a shared parametric sparseness prior for each θ_i ; a hyper-prior is placed on this shared parameter, and therefore the data from all M tasks are used to infer this parameter, resulting in a sharing of data/information across the M tasks. However, the multi-task algorithms discussed above assume all tasks are appropriate for sharing. In many practical applications one may anticipate that not all of the multiple CS measurements have sufficiently similar underlying signals, and therefore a component of the algorithm should cluster the multiple measurements appropriately. One may therefore anticipate an algorithm that first clusters the multiple CS measurements, and then within each cluster a multi-task CS algorithm of the type discussed above could be used. Rather than pursuing this two-step approach, here we simultaneously cluster the multiple CS measurements *and* perform the inversion of the underlying signals. In addition to inferring the multiple underlying signals, the algorithm also infers the appropriate sharing/clustering structure across the M tasks. Toward this end, we introduce a Dirichlet process (DP) prior [16] to the

hierarchical BCS model.

Dirichlet process priors allow uncertainty in the distributions chosen in a hierarchical model. In particular, suppose the distribution G is assigned a Dirichlet process prior, denote $G \sim DP(\lambda, G_0)$, where λ is a positive scalar precision parameter and G_0 is the base distribution. Then, there is a positive probability that a sample drawn from a DP will be as close as desired to any probability distribution having the same support as G_0 [17]. Therefore, the DP is rich enough to approximate any possible distribution, and hence has considerable advantage over parametric hierarchical models, which necessarily rely on very strong restrictions on distributional shape.

As detailed below, an important property of DP for the work presented here is that it provides a tool for non-parametric clustering (*i.e.*, the number of clusters need not be set in advance). The DP-based hierarchical model is employed to realize the desired property of simultaneously clustering and CS inversion of the M measurements $\{\mathbf{v}_i\}_{i=1,M}$. A variational Bayes [5] inference algorithm is considered, yielding a full posterior over the model parameters $\boldsymbol{\theta}_i$. Additionally, we also develop an efficient simplified VB DP algorithm that has good performance while significantly reducing computation time.

The remainder of the paper is organized as follows. The proposed DP multi-task compressive sensing framework is described in Section 2. Section 3 provides a variational Bayes inference algorithm, and a more-efficient simplified DP inference algorithm is developed in Section 4. In Section 5 we present experimental results, first on synthesized data to illustrate the underlying algorithmic mechanisms, and then on real image data. Section 6 concludes the work and outlines future research directions.

II. MULTI-TASK CS MODELING WITH DP PRIORS

A. Multi-Task CS Formulation for Global Sharing

Let \mathbf{v}_i represent the CS measurements associated with task i , and assume a total of M tasks. The i -th CS measurement may be represented as

$$\mathbf{v}_i = \Phi_i \boldsymbol{\theta}_i + \boldsymbol{\epsilon}_i, \quad (2)$$

where the CS measurements \mathbf{v}_i are characterized by an n_i -dimensional real vector, the sensing matrix Φ_i corresponding to task i is of size $n_i \times m$, and $\boldsymbol{\theta}_i$ is the set of (sparse) transform coefficients associated with task i . The j^{th} coefficient of $\boldsymbol{\theta}_i$ is denoted $\theta_{i,j}$. The residual error

vector $\epsilon_i \in \mathbb{R}^{n_i}$ is modeled as n_i *i.i.d.* draws from a zero-mean Gaussian distribution with an unknown precision α_0 (variance $1/\alpha_0$); the residual corresponds to the error imposed by setting the small transform coefficients exactly to zero when performing the CS inversion.

We impose a hierarchical sparseness prior on the parameters θ_i , the lower level of which is a zero-mean Gaussian distribution with a diagonal covariance matrix; the diagonal elements of the inverse covariance matrix are represented by the m -dimensional (precision) vector α_i . Specifically, we have

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

where $\alpha_{i,j}$ is the j^{th} component of the vector α_i . To impose sparseness, on a layer above a Gamma hyperprior is employed independently on the precisions $\alpha_{i,j}$, this yielding a Student-t distribution on $\theta_{i,j}$ when the precision parameters are integrated out [25] [26]; appropriate settings on the Gamma distributions yield a Student-t highly peaked around $\theta_{i,j} = 0$, constituting the desired sparseness promotion. The likelihood function for the parameters θ_i and α_0 , given the CS measurements \mathbf{v}_i , may be expressed as

$$p(\mathbf{v}_i | \theta_i, \alpha_0) = (2\pi/\alpha_0)^{-n_i/2} \exp(-\frac{\alpha_0}{2} \|\mathbf{v}_i - \Phi_i \theta_i\|_2^2). \quad (4)$$

Concerning the aforementioned hyperprior, for the multi-task CS model proposed in [23], the parameters $\alpha_i = \alpha$, for $i = 1, \dots, M$, and $\alpha \sim \prod_{j=1}^m Ga(c, d)$, which indicates that the parameters α are shared among all M tasks, and therefore the data from all CS measurements $\{\mathbf{v}_i\}_{i=1,M}$ contribute to learn the hyper-parameters and the precision $\alpha \sim Ga(a, b)$. In this framework the CS data from all M tasks are used to jointly infer the hyper-parameters α (global processing), and these shared hyper-parameters are then applied separately through the task-dependent likelihood function to infer the parameters of the specific task-dependent CS data (local processing). However, the assumption in such a setting, as well as related previous multi-task CS work [23][39][2], is that it is appropriate to employ all of the M tasks jointly to infer the hyper-parameters. One may envision problems for which the M tasks may be clustered into several sets of tasks (with the union of these sets constituting the M tasks), and data sharing may only be appropriate within each cluster. Through use of the Dirichlet process (DP) [38] employed below, we simultaneously cluster the multi-task CS data, and within each cluster the CS inversion is performed jointly. Consequently, we no longer need assume that all CS data from

the M tasks are appropriate for sharing (*i.e.*, rather than assuming $\alpha_i = \alpha$ for all $i = 1, \dots, M$, we will cluster the α_i , and each cluster will correspond to a specific class of sparseness for the associated transform coefficients θ_i).

B. Dirichlet Process for Clustered Sharing

The Dirichlet process, denoted as $DP(\lambda, G_0)$, is a measure on measures, and is parameterized by a positive scaling parameter λ and the base distribution G_0 . Assume we have $\{\alpha_i\}_{i=1,M}$ and each α_i is drawn identically from G , and G itself is a random measure drawn from a Dirichlet process,

$$\begin{aligned}\alpha_i | G &\stackrel{iid}{\sim} G, \quad i = 1, \dots, M, \\ G &\sim DP(\lambda, G_0),\end{aligned}$$

where the base distribution G_0 is a non-atomic base measure, and

$$E[G] = G_0. \quad (5)$$

The following important property of DP demonstrates how this prior can achieve parameter sharing. Defining $\alpha^{-i} = \{\alpha_1, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_M\}$ and integrating out G , the conditional distribution of α_i given α^{-i} follows a Pólya urn scheme of the form [4],

$$p(\alpha_i | \alpha^{-i}, \lambda, G_0) = \frac{\lambda}{\lambda + M - 1} G_0 + \frac{1}{\lambda + M - 1} \sum_{j=1, j \neq i}^M \delta_{\alpha_j}, \quad (6)$$

where δ_{α_j} denotes the distribution concentrated at single point α_j . Let $\{\tilde{\alpha}_k\}_{k=1}^{\tilde{K}}$ be the distinct values taken by $\{\alpha_i\}_{i=1,M}$ and let \tilde{n}_k^{-i} be the number of values in α^{-i} that equal $\tilde{\alpha}_k$. We can rewrite (6) as

$$p(\alpha_i | \alpha^{-i}, \lambda, G_0) = \frac{\lambda}{\lambda + M - 1} G_0 + \frac{1}{\lambda + M - 1} \sum_{k=1}^{\tilde{K}} \tilde{n}_k^{-i} \delta_{\tilde{\alpha}_k}. \quad (7)$$

Equation (7) shows that when considering α_i given all other observations α^{-i} , this new sample is either drawn from base distribution G_0 with probability $\frac{\lambda}{\lambda + M - 1}$, or is selected from the existing observed value $\tilde{\alpha}_k$ with probabilities proportional to the existing groups sizes \tilde{n}_k^{-i} . This property highlights the important sharing property of the Dirichlet process: a new sample prefers to join a group with a large population, *i.e.*, the more often a parameter is shared, the more likely it will be shared subsequently.

The scalar λ plays a balancing role between sampling a new parameter from the base distribution G_0 (“innovating”), or sharing previously sampled parameters. A larger λ yields more clusters, and in the limit $\lambda \rightarrow \infty$, $G \rightarrow G_0$ and we obtain a parametric hierarchical model; as $\lambda \rightarrow 0$, all $\{\alpha_i\}_{i=1,M}$ are aggregated into a single cluster and take on the same value. The framework is non-parametric in the sense that the number of clusters is not set explicitly, but is inferred from the data (the *expected* number of clusters is proportional to $\lambda \log M$ [16]).

The above DP representation underscores its sharing property, but without an explicit form for G . Sethuraman [30] provides an explicit characterization of G in terms of a stick-breaking construction. Consider two infinite collections of independent random variables π_k and α_k^* , $k = 1, 2, \dots, \infty$, where the π_k are drawn *i.i.d.* from a Beta distribution, denoted $Beta(1, \lambda)$, and the α_k^* are drawn *i.i.d.* from the base distribution G_0 . The stick-breaking representation of G is then defined as

$$G = \sum_{k=1}^{\infty} w_k \delta_{\alpha_k^*}, \quad (8)$$

with

$$w_k = \pi_k \prod_{i=1}^{k-1} (1 - \pi_i), \quad (9)$$

where

$$\begin{aligned} \pi_k | \lambda &\stackrel{iid}{\sim} Beta(1, \lambda), \\ \alpha_k^* | G_0 &\stackrel{iid}{\sim} G_0. \end{aligned}$$

This representation makes explicit that the random measure G is discrete with probability one and the support of G consists of an infinite set of atoms located at α_k^* , drawn independently from G_0 . The mixing weights w_k for atom α_k^* are given by successively breaking a unit length “stick” into an infinite number of pieces [30], with $0 \leq w_k \leq 1$ and $\sum_{k=1}^{\infty} w_k = 1$. In practice one also typically places a Gamma prior on λ [38].

The relationship between the stick-breaking representation and the Pólya urn scheme is interpreted as follows: if λ is large, each π_k drawn from $Beta(1, \lambda)$ will be very small, which means we will tend to have many sticks of very short length. Consequently, G will consist of an infinite number of α_k^* with very small weights w_k and therefore G will approach G_0 , the base distribution. For a small λ , each π_k drawn from $Beta(1, \lambda)$ will be large, which will result in a few large sticks with the remaining sticks very small. This leads to a clustering effect on the

parameters $\{\alpha_i\}_{i=1,M}$, as G will only have a large mass on a small subset of $\{\alpha_k^*\}_{k=1}^\infty$ (those α_k^* corresponding to the large sticks w_k). As discussed below, for the CS problem of interest here, we will impose a base distribution G_0 that encourages sparseness-promoting α_i .

C. Multi-Task CS with DP Priors

The base distribution G_0 corresponds to the sparseness promoting representation discussed in Section II A, and this yields the following hierarchical model for the CS measurements $\{\mathbf{v}_i\}_{i=1,M}$:

$$\begin{aligned}
\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0 &\sim \mathcal{N}(\Phi_i \boldsymbol{\theta}_i, \alpha_0^{-1} I), \quad i = 1, \dots, M, \\
\theta_{i,j} | \alpha_{i,j} &\sim \mathcal{N}(0, \alpha_{i,j}^{-1}), \quad j = 1, \dots, m, \quad i = 1, \dots, M, \\
\boldsymbol{\alpha}_i | G &\stackrel{iid}{\sim} G, \quad i = 1, \dots, M, \\
G | \lambda, c, d &\sim DP(\lambda, G_0), \\
G_0 &= \prod_{j=1}^m Ga(c, d), \\
\alpha_0 &\sim Ga(a, b), \\
\lambda &\sim Ga(e, f),
\end{aligned} \tag{10}$$

where $\alpha_{i,j}$ is the j -th element of $\boldsymbol{\alpha}_i$. Note that, if desired, we may also consider a separate precision α_0 for each of the tasks (*i.e.*, a task-dependent noise level), but here we assume it is the same across all tasks, for simplicity.

The choice of G_0 here is consistent with the sparseness-promoting hierarchical prior discussed in Section II-A. Addressing this further, consider task i and assume $\boldsymbol{\alpha}_i$ takes value $\boldsymbol{\alpha}_k^*$; the prior distribution over $\boldsymbol{\theta}_i$ is then

$$p(\boldsymbol{\theta}_i | c, d) = \prod_{j=1}^m \int \mathcal{N}(\theta_{i,j} | 0, \alpha_{k,j}^{*-1}) Ga(\alpha_{k,j}^* | c, d) d\alpha_{k,j}^*. \tag{11}$$

Equation (11) is a type of automatic relevance determination (ARD) prior which enforces the sparsity over $\boldsymbol{\theta}_i$ and has been utilized in sparse Bayesian learning such as the relevance vector machine (RVM) [34]. We usually set c and d very close to zero (*e.g.*, 10^{-4}) to make a broad prior over $\boldsymbol{\alpha}_k^*$ (here the product of Gamma distributions), which allows the posteriors on many of the elements of $\boldsymbol{\alpha}_k^*$ to concentrate at very large values, consequently the posteriors on the associated

elements of $\boldsymbol{\theta}_i$ concentrate at zero, and therefore the sparseness of $\boldsymbol{\theta}_i$ is achieved [25] [26]. Specifically, if $c = d = \nu/2$, (11) becomes a product of Student-t distributions, denoted as $t_\nu(0, 1)$ of degree ν . The Student-t distribution is known to have ‘‘heavy tails’’ compared to a Gaussian distribution, and allows for more robust shrinkage and borrowing of information.

To facilitate posterior computation, we employ a stick-breaking representation for the model in (10), which assumes that G has a form $G = \sum_{k=1}^{\infty} w_k \delta_{\boldsymbol{\alpha}_k^*}$. For the same purpose we also introduce an indicator variable z_i with $z_i = k$ indicating $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}_k^*$. Therefore the DP multi-task CS model is expressed as

$$\begin{aligned}
\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0 &\sim \mathcal{N}(\boldsymbol{\Phi}_i \boldsymbol{\theta}_i, \alpha_0^{-1} I), \quad i = 1, \dots, M, \\
\theta_{i,j} | z_i, \{\boldsymbol{\alpha}_k^*\}_{k=1,K} &\sim \mathcal{N}(0, \alpha_{z_i,j}^{*-1}), \quad j = 1, \dots, m, \quad i = 1, \dots, M, \\
z_i | \{w_k\}_{k=1,K} &\stackrel{iid}{\sim} \text{Multinomial}(\{w_k\}_{k=1,K}), \quad i = 1, \dots, M, \\
w_k &= \pi_k \prod_{l=1}^{k-1} (1 - \pi_l), \quad k = 1, \dots, K, \\
\pi_k &\stackrel{iid}{\sim} \text{Beta}(1, \lambda), \quad k = 1, \dots, K, \\
\lambda | e, f &\sim \text{Ga}(e, f), \\
\boldsymbol{\alpha}_k^* | c, d &\stackrel{iid}{\sim} \prod_{j=1}^m \text{Ga}(c, d), \quad i = 1, \dots, M, \\
\alpha_0 &\sim \text{Ga}(a, b),
\end{aligned} \tag{12}$$

where $1 \leq K \leq \infty$. For convenience, we denote the model in (12) as DP-MT CS. In practice K is chosen as a relatively large integer (e.g., $K = M$ if M is relatively large) which yields a negligible difference compared to the true DP [21], while making the computation practical. To understand the model better, a graphical representation corresponding to (10) and (12) is shown in Figure 1. Similarly, hyper-parameters a , b , e , and f are all set to a small value to have a non-informative prior over α_0 and λ respectively.

III. VARIATIONAL BAYESIAN INFERENCE

Given the DP-MT CS model framework and the prior distributions described in Section II, we require an inference algorithm to estimate the corresponding posterior distributions. One may perform inference via MCMC [18], however this requires vast computational resources and MCMC convergence is often difficult to diagnose [18]. Variational Bayes inference is therefore

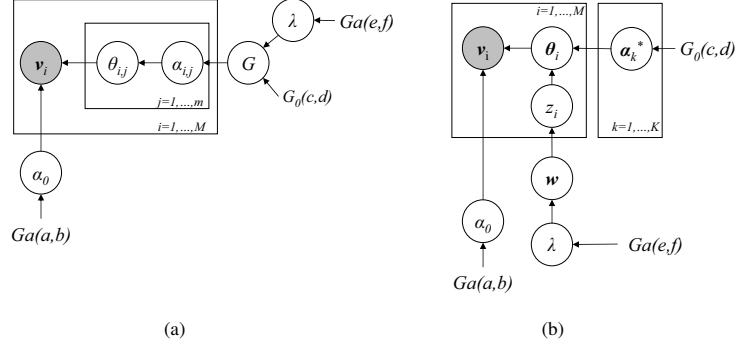


Fig. 1

GRAPHICAL REPRESENTATION OF THE DP MULTI-TASK CS. (A) IN A GENERAL DP FORM CORRESPONDING TO THE MODEL IN (10). (B) IN A STICK-BREAKING REPRESENTATION, CORRESPONDING TO THE MODEL IN (12).

introduced as a relatively efficient method for approximating the posterior. From Bayes' rule, we have

$$p(\mathbf{H}|\mathbf{V}, \Upsilon) = \frac{p(\mathbf{V}|\mathbf{H})p(\mathbf{H}|\Upsilon)}{\int p(\mathbf{V}|\mathbf{H})p(\mathbf{H}|\Upsilon)d\mathbf{H}}, \quad (13)$$

where $\mathbf{V} = \{\mathbf{v}_i\}_{i=1,M}$ are CS measurements from M CS tasks, $\mathbf{H} = \{\alpha_0, \lambda, \boldsymbol{\pi}, \{z_i\}_{i=1,M}, \{\boldsymbol{\theta}_i\}_{i=1,M}, \{\boldsymbol{\alpha}_k^*\}_{k=1,K}\}$ are hidden variables in the model (with $\boldsymbol{\pi} = \{\pi_k\}_{k=1,K}$) and $\Upsilon = \{a, b, c, d, e, f\}$ are hyper-parameters which determine the distributions of the hidden variables. The integration in the denominator of (13), called the *marginal likelihood*, or “evidence” [3], is generally intractable to compute analytically. Instead of directly estimating $p(\mathbf{H}|\mathbf{V}, \Upsilon)$, variational methods seek a distribution $q(\mathbf{H})$ to approximate the true posterior distribution $p(\mathbf{H}|\mathbf{X}, \Upsilon)$. Consider the log marginal likelihood

$$\log p(\mathbf{V}|\Upsilon) = \mathcal{F}(q(\mathbf{H})) + \mathcal{D}_{KL}(q(\mathbf{H})||p(\mathbf{H}|\mathbf{V}, \Upsilon)), \quad (14)$$

where

$$\mathcal{F}(q(\mathbf{H})) = \int q(\mathbf{H}) \log \frac{p(\mathbf{V}|\Upsilon)p(\mathbf{H}|\Upsilon)}{q(\mathbf{H})} d\mathbf{H}, \quad (15)$$

and

$$\mathcal{D}_{KL}(q(\mathbf{H})||p(\mathbf{H}|\mathbf{V}, \Upsilon)) = \int q(\mathbf{H}) \log \frac{q(\mathbf{H})}{p(\mathbf{H}|\mathbf{V}, \Upsilon)} d\mathbf{H}. \quad (16)$$

The expression $\mathcal{D}_{KL}(q(\mathbf{H})||p(\mathbf{H}|\mathbf{V}, \Upsilon))$ is the KL divergence between the approximate $q(\mathbf{H})$ and true posterior $p(\mathbf{H}|\mathbf{V}, \Upsilon)$. The approximation of the true posterior $p(\mathbf{H}|\mathbf{V}, \Upsilon)$ using $q(\mathbf{H})$ can be achieved by minimizing $\mathcal{D}_{KL}(q(\mathbf{H})||p(\mathbf{H}|\mathbf{V}, \Upsilon))$. Since the KL divergence is nonnegative

and $\log p(\mathbf{V}|\Upsilon)$ is fixed given \mathbf{V} , from (14) this minimization is equivalent to maximization of $\mathcal{F}(q(\mathbf{H}))$, which forms a strict lower bound on $\log p(\mathbf{V}|\Upsilon)$,

$$\log p(\mathbf{V}|\Upsilon) \geq \mathcal{F}(q). \quad (17)$$

Therefore estimation of $q(\mathbf{H})$ is transformed from minimizing $\mathcal{D}_{KL}(q(\mathbf{H})||p(\mathbf{H}|\mathbf{V}, \Upsilon))$ to maximizing $\mathcal{F}(q)$, which may be made computationally tractable. In particular, for computational convenience, $q(\mathbf{H})$ is expressed in a factorized form, with the same functional form as the priors $p(\mathbf{H}|\Upsilon)$. For the model in (12), we assume

$$q(\mathbf{H}) = q(\alpha_0)q(\lambda)q(\boldsymbol{\pi}) \prod_{i=1}^M q(z_i) \prod_{i=1}^M q(\boldsymbol{\theta}_i) \prod_{k=1}^K q(\boldsymbol{\alpha}_k^*), \quad (18)$$

where

$$\begin{aligned} q(\alpha_0) &\sim Ga(\tilde{a}, \tilde{b}), \\ q(\lambda) &\sim Ga(\tilde{e}, \tilde{f}), \\ q(\boldsymbol{\pi}) &\sim \prod_{k=1}^{K-1} Beta(\tau_{1k}, \tau_{2k}), \\ q(z_i) &\sim Multinomial(\mathbf{w}), \quad i = 1, \dots, M, \\ q(\boldsymbol{\theta}_i) &\sim \mathcal{N}(\mu_i, \boldsymbol{\Gamma}_i), \quad i = 1, \dots, M, \\ q(\boldsymbol{\alpha}_k^*) &\sim \prod_{j=1}^m Ga(\alpha_{k,j}^* | \tilde{c}_{k,j}, \tilde{d}_{k,j}), \quad k = 1, \dots, K, \end{aligned} \quad (19)$$

where $\mathbf{w} = \{w_k\}_{k=1, K}$.

The joint distribution of \mathbf{H} and observations \mathbf{V} are given as

$$\begin{aligned} p(\mathbf{H}, \mathbf{V}|\Upsilon) &= p(\alpha_0|a, b)p(\lambda|e, f)p(\boldsymbol{\pi}|\lambda) \prod_{i=1}^M p(z_i|\mathbf{w}) \cdot \\ &\cdot \prod_{i=1}^M p(\boldsymbol{\theta}_i|z_i, \{\boldsymbol{\alpha}_k^*\}_{k=1, K}) \prod_{k=1}^K p(\boldsymbol{\alpha}_k^*|c, d) \prod_{i=1}^M p(\mathbf{v}_i|\boldsymbol{\theta}_i, \alpha_0), \end{aligned} \quad (20)$$

where all these prior distributions are given in (12). All hyper-parameters Υ in these priors distributions are assumed to be fixed in advance.

By substituting (18) and (20) into (15), the lower bound $\mathcal{F}(q)$ is readily obtained. The optimization of the lower bound $\mathcal{F}(q)$ is realized by taking functional derivatives with respect to each of the $q(\cdot)$ distributions while fixing the other q distributions, and setting $\partial\mathcal{F}(q)/\partial q(\cdot) = 0$ to

find the distribution $q(\cdot)$ that increases \mathcal{F} [3]. The update equations for the variational posteriors are summarized in the Appendix. The convergence of the algorithm is monitored by the increase of the lower bound \mathcal{F} ; once the increase of \mathcal{F} is below a preset small value, we terminate the algorithm. One practical issue of the variational Bayesian inference is that the VB algorithm converges to a local maximum of the lower bound of the marginal log-likelihood since the true posterior usually is multi-modal. Therefore the average of multiple runs of the algorithm from different starting points may avoid this issue and yield better performance.

IV. SIMPLIFIED DP MULTI-TASK CS

Although VB-based DP-MT CS performs well, as demonstrated below when presenting experimental results, the computation is relatively expensive. In order to develop an efficient algorithm for multi-task CS, we simplify the DP-MT CS model by directly estimating the values of α_i rather than yielding a full posterior, as discussed in Section II-C and III.

Toward this end, we adopt an alternative finite approximation to the DP, in which we assume α_i is drawn from a mixture model

$$p(\alpha_i | \{l_k\}_{k=1,J}, \{\alpha_k^*\}_{k=1,J}) = \sum_{k=1}^J l_k \delta_{\alpha_k^*}, \quad (21)$$

where $\sum_{k=1}^J l_k = 1$ and $\delta_{\alpha_k^*}$ denotes the distribution concentrated at a single point α_k^* . Equation (21) indicates that (i) α_i has a probability l_k of taking the value α_k^* ; and (ii) when α_i and α_j take the same value α_k^* , task i and j share parameters (and associated CS data). We place a Dirichlet prior $Dir(1/J, \dots, 1/J)$ over the mixing weights $\{l_k\}_{k=1,J}$. This model is a finite approximation to the DP-based model, in the sense that as $J \rightarrow \infty$ this prior corresponds to DP with $\lambda = 1$ [30][22]. Finally, a set of indicator variables $\{z_i\}_{i=1,M}$ are introduced, with $z_i = k$

if $\alpha_i = \alpha_k^*$. The overall model for (2) is therefore summarized as

$$\begin{aligned}
\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0 &\sim \mathcal{N}(\Phi_i \boldsymbol{\theta}_i, \alpha_0^{-1} \mathbf{I}), \quad i = 1, \dots, M, \\
\theta_{i,j} | \alpha_{z_i,j}^* &\sim \mathcal{N}(0, \alpha_{z_i,j}^{*-1}), \quad j = 1, \dots, m, \quad i = 1, \dots, M, \\
\boldsymbol{\alpha}_i | \{l_k\}_{k=1,J}, \{\alpha_k^*\}_{k=1,J} &\stackrel{iid}{\sim} \sum_{k=1}^J l_k \delta_{\alpha_k^*} \quad i = 1, \dots, M, \\
z_i &\stackrel{iid}{\sim} \text{Multinomial}(\{l_k\}_{k=1,J}), \quad i = 1, \dots, M, \\
\{l_k\}_{k=1,J} &\sim \text{Dir}(1/J, \dots, 1/J), \\
\alpha_0 &\sim \text{Ga}(a, b).
\end{aligned} \tag{22}$$

The model in (22) is termed a simplified DP multi-task CS algorithm, because we directly estimate the values of α_k^* instead of the full posterior distributions as in DP-MT CS. However, a (Gaussian) posterior is still provided for the transform coefficients $\boldsymbol{\theta}_i$; in this sense the model is related to the manner in which Tipping performed inference with the relevance vector machine [33], although here we are considering a multi-task scenario. The likelihood function for $\boldsymbol{\theta}_i$ based on the CS measurements \mathbf{v}_i , given $\boldsymbol{\alpha}_i$ and α_0 , is a multivariate Gaussian distribution

$$\begin{aligned}
p(\boldsymbol{\theta}_i | \mathbf{v}_i, \boldsymbol{\alpha}_i, \alpha_0) &= \frac{p(\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0) p(\boldsymbol{\theta}_i | \boldsymbol{\alpha}_i)}{\int p(\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0) p(\boldsymbol{\theta}_i | \boldsymbol{\alpha}_i) d\boldsymbol{\theta}_i} \\
&= \mathcal{N}(\boldsymbol{\theta}_i | \alpha_0 \boldsymbol{\Sigma}_i \Phi_i^T \mathbf{v}_i, \boldsymbol{\Sigma}_i),
\end{aligned} \tag{23}$$

where $\boldsymbol{\Sigma}_i = (\alpha_0 \Phi_i^T \Phi_i + \mathbf{A})^{-1}$ and $\mathbf{A} = \text{diag}(\boldsymbol{\alpha}_i)$ is a diagonal matrix with the diagonal elements are the elements of the vector $\boldsymbol{\alpha}_i$.

This model can be solved to give a point estimate of α_k^* and α_0 . However before we proceed to that step, we notice that what we are really interested in are α_k^* rather than α_0 , and that the model estimation is very sensitive to the initial guess of α_0 [23]; integrating out the noise precision α_0 collapses the model to a lower dimensional space and also allows efficient sequential

optimization [23][20]. We adopt the modified formalism [23]

$$\begin{aligned}
\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0 &\sim \mathcal{N}(\Phi_i \boldsymbol{\theta}_i, \alpha_0^{-1} \mathbf{I}), \quad i = 1, \dots, M, \\
\theta_{i,j} | \alpha_{i,j} &\sim \mathcal{N}(0, \alpha_0^{-1} \alpha_{i,j}^{-1}), \quad j = 1, \dots, m, \quad i = 1, \dots, M, \\
\alpha_0 | \nu &\sim \text{Ga}\left(\frac{\nu}{2}, \frac{\nu}{2}\right), \\
\boldsymbol{\alpha}_i | \{l_k\}_{k=1,J}, \{\boldsymbol{\alpha}_k^*\}_{k=1,J} &\stackrel{iid}{\sim} \sum_{k=1}^J l_k \delta_{\boldsymbol{\alpha}_k^*} \quad i = 1, \dots, M, \\
z_i &\stackrel{iid}{\sim} \text{Multinomial}(\{l_k\}_{k=1,J}), \quad i = 1, \dots, M, \\
\{l_k\}_{k=1,J} &\stackrel{iid}{\sim} \text{Dir}(1/J, \dots, 1/J).
\end{aligned} \tag{24}$$

Although the noise variance α_0 still appears in the model, which helps to understand its structure, it is eventually integrated out analytically when performing inference. We denote this modified model as SimDP-MT CS for convenience. This representation changes the likelihood function for $\boldsymbol{\theta}_i$ from a multivariate Gaussian distribution in (23) to a multivariate t -distribution given \mathbf{v}_i and $\boldsymbol{\alpha}_i$ [23], which may be expressed as

$$\begin{aligned}
p(\boldsymbol{\theta}_i | \mathbf{v}_i, \boldsymbol{\alpha}_i) &= \int p(\boldsymbol{\theta}_i | \mathbf{v}_i, \boldsymbol{\alpha}_i, \alpha_0) p(\alpha_0 | a, b) d\alpha_0 \\
&= \frac{\Gamma(\frac{\nu+m}{2}) [1 + \frac{1}{\nu} (\boldsymbol{\theta}_i - \tilde{\boldsymbol{\mu}}_i)^T \tilde{\boldsymbol{\Sigma}}_i^{-1} (\boldsymbol{\theta}_i - \tilde{\boldsymbol{\mu}}_i)]^{-\frac{\nu+m}{2}}}{\Gamma(\frac{\nu}{2}) (\nu)^{\frac{m}{2}} |\tilde{\boldsymbol{\Sigma}}_i|^{\frac{1}{2}}},
\end{aligned} \tag{25}$$

with

$$\begin{aligned}
\tilde{\boldsymbol{\mu}}_i &= \tilde{\boldsymbol{\Sigma}}_i \Phi_i^T \mathbf{v}_i, \\
\tilde{\boldsymbol{\Sigma}}_i &= (\Phi_i^T \Phi_i + \mathbf{A})^{-1},
\end{aligned} \tag{26}$$

where $\mathbf{A} = \text{diag}(\boldsymbol{\alpha}_{i,1}, \dots, \boldsymbol{\alpha}_{i,m})$. The likelihood function in (25) is a multivariate t -distribution $t_\nu(\tilde{\boldsymbol{\mu}}_i, \tilde{\boldsymbol{\Sigma}}_i)$ with ν degrees of freedom, which induces a heavy-tailed distribution on the basis coefficients $\boldsymbol{\theta}_i$, the heavy tail for which allows more robust shrinkage and information sharing among tasks.

The SimDP-MT CS model, integrating out α_0 in (24), can be solved through the variational methods similar to the VB inference in Section II. We use a variational posterior $q(\{z_i\}_{i=1,M}, \{l_k\}_{k=1,J})$ to approximate the true posterior $p(\{z_i\}_{i=1,M}, \{l_k\}_{k=1,J} | \{\mathbf{v}_i\}_{i=1,M})$ and assume a

factorized form for the variational posterior

$$q(\{z_i\}_{i=1,M}, \{l_k\}_{k=1,J}) = \prod_{i=1}^M q(z_i)q(\{l_k\}_{k=1,J}), \quad (27)$$

where $q(z_i) = \text{Multinomial}(\{l_k\}_{k=1,J})$ and $q(\{l_k\}_{k=1,J}) = \text{Dir}(\{\omega_k\}_{k=1,J})$.

The lower bound of the marginal log likelihood is written as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}^*, \boldsymbol{\omega}) &= \int \int q(\mathbf{z}, \mathbf{l}) \cdot [\log p(\mathbf{v}, \mathbf{z}, \mathbf{l} | \boldsymbol{\alpha}^*, \boldsymbol{\omega}) - \log q(\mathbf{z}, \mathbf{l})] d\mathbf{z} d\mathbf{l} \\ &= \int \int q(\mathbf{l}) \prod_{i=1}^M q(z_i) [\log p(\mathbf{l} | \boldsymbol{\omega}) + \sum_{i=1}^M \log p(z_i | \mathbf{l}) + \log p(\mathbf{v}_i | \boldsymbol{\alpha}_{z_i}^*) - \log q(\mathbf{l}) \\ &\quad - \sum_{i=1}^M \log q(z_i)] d\mathbf{z} d\mathbf{l} \end{aligned} \quad (28)$$

with $\boldsymbol{\alpha}^* = \{\boldsymbol{\alpha}_k^*\}_{k=1,J}$, $\boldsymbol{\omega} = \{\omega_k\}_{k=1,J}$, $\mathbf{l} = \{l_k\}_{k=1,J}$, and $\mathbf{z} = \{z_i\}_{i=1,M}$.

Estimation of $\boldsymbol{\alpha}^*$ and $\boldsymbol{\omega}$ can be obtained by maximizing the lower bound $\mathcal{L}(\boldsymbol{\alpha}^*, \boldsymbol{\omega})$ in (28) via the expectation-maximization (EM) algorithm [11]. In the E-step $\boldsymbol{\omega}$ is estimated by maximizing $\mathcal{L}(\boldsymbol{\alpha}^*, \boldsymbol{\omega})$ given $\boldsymbol{\alpha}^*$ as the most current estimated values. Specifically, $q(\mathbf{l})$ and $q(\mathbf{z})$ are updated separately by maximizing the lower bound given other $q(\cdot)$ and $\boldsymbol{\alpha}^*$; the updating equations are presented in Appendix II. In the M-step, values of $\boldsymbol{\alpha}^*$ are estimated by maximizing (28) given $q(\mathbf{l})$ and $q(\mathbf{z})$ (*i.e.*, $\boldsymbol{\omega}$). Let $\kappa_{i,k} = q(z_i = k)$ and then (28) becomes

$$\mathcal{L}(\boldsymbol{\alpha}^*) = \sum_{k=1}^J \mathcal{L}_k(\boldsymbol{\alpha}_k^*), \quad (29)$$

where

$$\begin{aligned} \mathcal{L}_k(\boldsymbol{\alpha}_k^*) &= \sum_{i=1}^M \kappa_{i,k} \log p(\mathbf{v}_i | \boldsymbol{\alpha}_k^*) \\ &= \sum_{i=1}^M \kappa_{i,k} \log \int p(\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0) p(\boldsymbol{\theta}_i | \boldsymbol{\alpha}_k^*, \alpha_0) p(\alpha_0 | \nu) d\boldsymbol{\theta}_i d\alpha_0, \\ &= -\frac{1}{2} \sum_{i=1}^M \kappa_{i,k} \left[(n_i + \nu) \log(\mathbf{v}_i^T \mathbf{B}_{i,k}^{-1} \mathbf{v}_i + \frac{\nu}{2}) + \log |\mathbf{B}_{i,k}| \right] + \text{const}, \end{aligned} \quad (30)$$

with $\mathbf{B}_{i,k} = \mathbf{I} + \boldsymbol{\Phi}_i \mathbf{A}_k^{-1} \boldsymbol{\Phi}_i^T$ and $\mathbf{A}_k = \text{diag}(\{\alpha_{k,j}^*\}_{j=1,m})$; $\boldsymbol{\alpha}_k^*$ is obtained by maximizing $\mathcal{L}_k(\boldsymbol{\alpha}_k^*)$.

Equation (30) may be maximized by direct differentiation, however the computation complexity is similar to the VB approach for the DP-MT CS. A modified fast algorithm to maximize

(30) may be derived based on the fast algorithm in [23], which is developed in a manner similar to the fast RVM algorithm [33]. Given that $\mathbf{B}_{i,k}$ in (30) can be decomposed as

$$\begin{aligned}\mathbf{B}_{i,k} &= \mathbf{I} + \sum_{t=1, t \neq j}^m \alpha_{k,t}^*{}^{-1} \Phi_{i,t} \Phi_{i,t}^T + \alpha_{k,j}^*{}^{-1} \Phi_{i,j} \Phi_{i,j}^T \\ &= \mathbf{B}_{i,k,-j} + \alpha_{k,j}^*{}^{-1} \Phi_{i,j} \Phi_{i,j}^T,\end{aligned}\quad (31)$$

where $j \in \{1, 2, \dots, m\}$ and $\mathbf{B}_{i,k,-j}$ is obtained by removing the contribution of $\alpha_{k,j}^*{}^{-1}$, the j^{th} component in α_k^* , as well as that of $\Phi_{i,j}$, the j^{th} column in Φ_i , from $\mathbf{B}_{i,k}$. From the matrix determinant and inverse identities we have

$$|\mathbf{B}_{i,k}| = |\mathbf{B}_{i,k,-j}| |1 + \alpha_{k,j}^*{}^{-1} \Phi_{i,j}^T \mathbf{B}_{i,k,-j}^{-1} \Phi_{i,j}|, \quad (32)$$

$$\mathbf{B}_{i,k}^{-1} = \mathbf{B}_{i,k,-j}^{-1} - \frac{\mathbf{B}_{i,k,-j}^{-1} \Phi_{i,j} \Phi_{i,j}^T \mathbf{B}_{i,k,-j}^{-1}}{\alpha_{k,j}^* \mathbf{B}_{i,k,-j}^{-1} \Phi_{i,j}}. \quad (33)$$

Substituting (32) and (33) into (30), we can decompose $\mathcal{L}_k(\alpha_k^*)$ into two independent parts:

$$\begin{aligned}\mathcal{L}_k(\alpha_k^*) &= -\frac{1}{2} \sum_{i=1}^M \kappa_{i,k} \left[(n_i + \nu) \log(\mathbf{v}_i^T \mathbf{B}_{i,k}^{-1} \mathbf{v}_i + \nu) + \log |\mathbf{B}_{i,k}| \right] + \text{const} - \\ &\quad - \frac{1}{2} \sum_{i=1}^M \kappa_{i,k} \left[\log(1 + \alpha_{k,j}^*{}^{-1} s_{i,k,j}) + (n_i + \nu) \log \left(1 - \frac{q_{i,k,j}^2 / g_{i,k,j}}{\alpha_{k,j}^* + s_{i,k,j}} \right) \right] \\ &= \mathcal{L}_k(\alpha_{k,-j}^*) + L_k(\alpha_{k,j}^*),\end{aligned}\quad (34)$$

where $\alpha_{k,-j}^*$ is α_k^* with the j^{th} component $\alpha_{k,j}^*$ is removed and

$$s_{i,k,j} \triangleq \Phi_{i,j}^T \mathbf{B}_{i,k,-j}^{-1} \Phi_{i,j}, \quad q_{i,k,j} \triangleq \Phi_{i,j}^T \mathbf{B}_{i,k,-j}^{-1} \mathbf{v}_i, \quad \text{and} \quad g_{i,k,j} \triangleq \mathbf{v}_i^T \mathbf{B}_{i,k,-j}^{-1} \mathbf{v}_i + \nu. \quad (35)$$

Equation (34) indicates that the dependence of $\mathcal{L}_k(\alpha_k^*)$ on a single parameter $\alpha_{k,j}^*$ can be isolated from all the other parameters $\alpha_{k,-j}^*$. An increase of $\mathcal{L}_k(\alpha_k^*)$ can be achieved by sequentially maximizing $L_k(\alpha_{k,j}^*)$ for varying j ,

$$\begin{aligned}\alpha_{k,j}^* &= \arg \max L_k(\alpha_{k,j}^*) \\ &\approx \frac{\sum_{i=1}^M \kappa_{i,k}}{\sum_{i=1}^M \kappa_{i,k} \frac{(n_i + \nu) q_{i,k,j}^2 / g_{i,k,j} - s_{i,k,j}}{s_{i,k,j} (s_{i,k,j} - q_{i,k,j}^2 / g_{i,k,j})}}, \quad \text{if } \sum_{i=1}^M \kappa_{i,k} \frac{(n_i + \nu) q_{i,k,j}^2 / g_{i,k,j} - s_{i,k,j}}{s_{i,k,j} (s_{i,k,j} - q_{i,k,j}^2 / g_{i,k,j})} > 0, \\ \alpha_{k,j}^* &= \infty, \quad \text{otherwise.}\end{aligned}\quad (36)$$

The approximation of $\alpha_{k,j}^*$ is from the assumption that $\alpha_{k,j}^* \ll s_{i,k,j}$ since in practice we found this to be true for all cases considered. Assuming $z_i = k$, since $\alpha_{k,j}^*$ is a precision parameter over

$\theta_{i,j}$, therefore $\alpha_{k,j}^* = \infty$ is equivalent to $\theta_{i,j} = 0$, which implies that we only need to estimate nonzero $\theta_{i,j}$ and let $\alpha_{k,j}^*$ control which $\theta_{i,j}$'s are nonzero. At the same time, if $\alpha_{k,j}^* = \infty$, *i.e.*, $\theta_{i,j} = 0$, column vector $\Phi_{i,j}$ can be removed from the representation; otherwise, $\Phi_{i,j}$ is added. Therefore the term $\sum_{i=1}^M \kappa_{i,k} \frac{(n_i + \nu) q_{i,k,j}^2 / g_{i,k,j}^{-s_{i,k,j}}}{s_{i,k,j} (s_{i,k,j} - q_{i,k,j}^2 / g_{i,k,j})}$, denoted as $\zeta_{k,j}$, acts as a controller to determine the presence of $\Phi_{i,j}$ for k^{th} mixture component α_k^* . The values of $s_{i,k,j}$, $q_{i,k,j}$ and $g_{i,k,j}$ are computed efficiently where only currently presented columns of Φ_i are considered.

A framework for efficient inference is described as follows:

- 1) Initialize $\kappa_{i,k}$ for $k = 1, \dots, J, i = 1, \dots, M$; for each k select a candidate basis $\Phi_{i,j}$ and then initialize $\alpha_{k,j}^*$, $k = 1, \dots, J, i = 1, \dots, M$.
- 2) Compute the task membership $\kappa_{i,k}$ for $k = 1, \dots, J, i = 1, \dots, M$.
- 3) For $k = 1, \dots, J$, select a candidate basis $\Phi_{i,j}$ (here we choose one that maximizes $\mathcal{L}_k(\alpha_k^*)$ in (34)) and update $\alpha_{k,j}^*$ (add/delete/re-estimate) according to the value of $\zeta_{k,j}$; compute $\tilde{\mu}_{i,k}$ and $\tilde{\Sigma}_{i,k}$ in (26) in which Φ_i has basis functions having zero coefficients excluded.
- 4) Compute algorithm terminating criterion, which is based on the lower bound in (28) being below a preset threshold. If it is below a preset threshold, then stop, otherwise go back to step 2).

Step 3) is very similar to the fast algorithm proposed in [23] and essentially our fast algorithm consists of two loops: (i) the outer loop includes step 2) and 3) and computes the membership for each task given a particular mixture component (clustering), and (ii) the inner loop including step 3) computes the basis updating for each component given all tasks (cluster-dependent updating of model parameters).

We note that to date the convergence of the above efficient algorithm is not theoretically guaranteed. In the fast RVM algorithm [33], the observations are fixed for all iterations when updating basis, however in the above algorithm, the observations may vary for each iteration (essentially the contribution from each CS task may vary). Therefore, the lower bound may not monotonically increase at the beginning iterations. However, from our practical experience, the task membership stabilizes very quickly (usually within 100 iterations). This is reasonable since we first choose those bases that increase the lower bound most, which indicates that these bases are most informative and able to tell the inter-task relationship. Once the task membership remains unchanged for each iteration, then the basis updating for each α_k^* monotonically increases the lower bound. Therefore we still choose the increase of the lower bound as the algorithm

stop criterion, and from our experiments we observe that the efficient algorithm works well.

One may observe that the basic models are different for the DP-MT CS in (12) and the SimDP-MT CS in (24), where we need to estimate the posterior distribution of α_0 in the former while in the latter we integrate out α_0 (α_0 does not appear in the inference and therefore no estimation is required for α_0). Specifically, in (12) the prior distribution for θ_i is independent of α_0 while in (24) it is dependent of α_0 . The reason that we do not integrate out α_0 in (12) is for the purpose of inference convenience (conjugacy). If we choose the basic model for (12) as in (24), then when considering the posterior distribution for α_k^* (assuming $\alpha_i = \alpha_k^*$), the corresponding likelihood function for α_k^* is a multivariate t -distribution as in (23), to which the Gamma prior over α_k^* is not conjugate. This issue does not exist in the SimDP-MT CS in (24), because there we directly estimate the values of α_k^* rather than the posterior distribution and therefore no conjugacy is required. Since in (12) we estimate a posterior on α_0 rather than a point estimate (as in the SimDP-MT solution), it is anticipated that this model will perform well, albeit at greater computational cost relative to the SimDP-MT representation in (24).

V. EXPERIMENTAL RESULTS

A. Synthetic data

In the first set of examples we consider synthesized data, with the objective of examining the sharing mechanisms associated with the DP-based multi-task CS inversion. These simple and illustrative examples allow us to investigate how signal sparseness impacts the sharing mechanisms, and to determine when simpler global sharing [23], [2], [39] is appropriate. Data of the form considered here is presented in Figure 2. Specifically, in separate examples we generate data in which it is anticipated that there are 10, 5, 3, 2 and 1 underlying clusters, and the data are generated such that five different data vectors are associated with each cluster. First consider Figure 2, which corresponds to the ten-cluster case. The sparse signals in Figure 2 correspond to ten “templates”, each corresponding to a 256-length signal, with 30 non-zero components (the values of those non-zero components are randomly drawn from a zero-mean Gaussian with unit variance). The non-zero locations are chosen randomly for each signal such that the correlation between these sparse templates is zero. For each of these ten templates, five sparse signals are generated, each with 256 samples (50 total signals or tasks are generated); the five cluster-specific sparse signals are generated by randomly selecting three non-zero elements

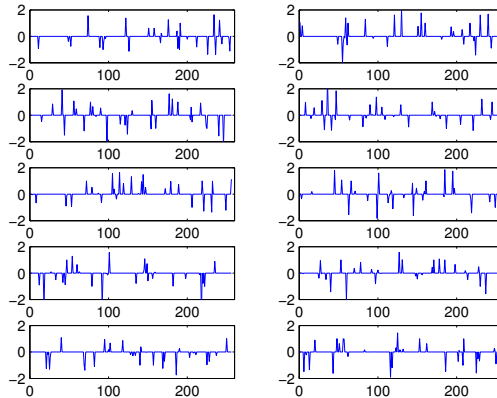


Fig. 2

TEN TEMPLATE SIGNALS FOR 10-CLUSTER CASE.

from the associated template and setting the coefficients to zero, and three zero-amplitude points in the template are randomly now set to be non-zero (each of these three non-zero values again drawn from $\mathcal{N}(0, 1)$). In this manner the sparseness properties of the five signals generated from a given template are highly related, and the ten clusters of sparse signals (each cluster composed of five signals) have distinct sparseness properties. For each sparse signal a set of CS random projections are performed, with the components of each projection vector drawn randomly from a zero-mean, unit-variance Gaussian random variable. In Figure 3 are shown the reconstruction errors of the CS inversion, as a function of the number of CS measurements. In Figure 3 two sets of results are presented, both based on variational Bayesian inference (the inference method is the same, and therefore only the models are distinct); in the DP-MT formulation the DP-based prior is employed, while in the MT* results a shared sparseness prior is employed (as in [23], across all 50 CS measurements). From Figure 3 the advantage of the DP-based formulation is evident. The VB DP-MT algorithm is initialized by setting the hyper-parameters $a = 10^{-4}$, $b = 10^{-3}\text{std}(\mathbf{v})^2$ (std denotes the standard deviation), $c = d = e = f = 10^{-4}$; the membership $\kappa_{i,k}$ for task i is set randomly, followed by normalization. The experiment was run 100 times (with 100 different random generations of random projection as well as initial membership), and the error bars in Figure 3 represent the standard deviation about the mean.

It is also of interest to examine the number of clusters that are inferred by the DP-based multi-task CS inversion algorithm. In Figure 3 we also present histograms for the number of

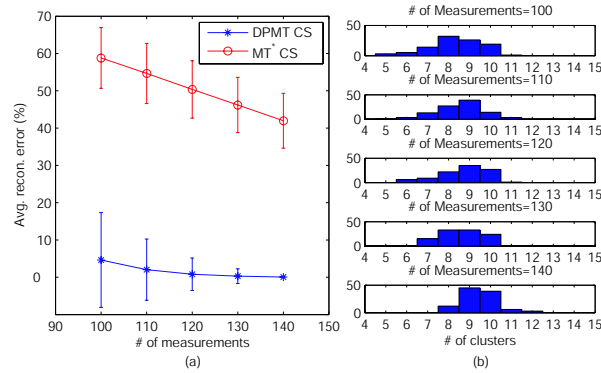


Fig. 3

MULTI-TASK CS INVERSION ERROR (IN PERCENT) FOR DP-BASED INVERSION AND WHEN A SHARED SPARSENESS PRIOR IS CONSIDERED. THESE RESULTS ARE FOR THE CASE IN WHICH THE DATA WERE GENERATED BASED ON TEN CLUSTERS, WITH FIVE SPARSE SIGNALS GENERATED RANDOMLY FOR EACH OF THE TEN TEMPLATES/CLUSTERS (SEE FIGURE 2). RESULTS ARE SHOWN FOR 100 DIFFERENT RANDOM GENERATIONS OF THE DATA. (A) RECONSTRUCTION ERRORS, (B) HISTOGRAMS OF THE NUMBER OF CLUSTERS YIELDED BY DP-MT.

different clusters inferred by the DP analysis, recalling that the data were generated with the idea of constituting 10 clusters. It is clear from Figure 3 that the algorithm tends to infer about 10 clusters, but there is some variation, with the variation in the number of clusters increasing with decreasing number of CS measurements. This distribution on the number of clusters yields interesting insight concerning CS-based multi-task inversion, with this revisited below.

As a comparison to the two multi-task results in Figure 3, in Table I we compare the average error and standard deviation of the multi-task algorithms with single-task CS inversion (each CS inversion performed separately for each CS measurement, again based on a variational Bayesian inference formulation). Note that the advantage of DP-based multi-task inversion becomes more

Measurements	60	70	80	90	100	110	120	130
DP-MT	58.79(25.49)	42.65(30.01)	24.87(28.11)	7.95(13.41)	4.63(12.72)	2.04(8.21)	0.81(4.35)	0.30(1.97)
MT*	77.07(9.15)	72.49(8.93)	67.73(8.22)	63.18(8.75)	58.81(8.13)	54.65(8.07)	50.37(7.70)	46.20(7.43)
ST	92.91(25.38)	65.03(37.79)	25.37(35.41)	12.57(17.63)	7.75(18.91)	3.39(7.91)	2.49(2.68)	1.69(0.41)

TABLE I

AVERAGE RECONSTRUCTION ERROR (%) WITH STD IN THE BRACKETS; RESULTS ARE FOR DATA GENERATION BASED ON TEMPLATES, AND A TOTAL OF 50 “TASKS”, FIVE SYNTHESIZED FOR EACH OF THE TEMPLATES.

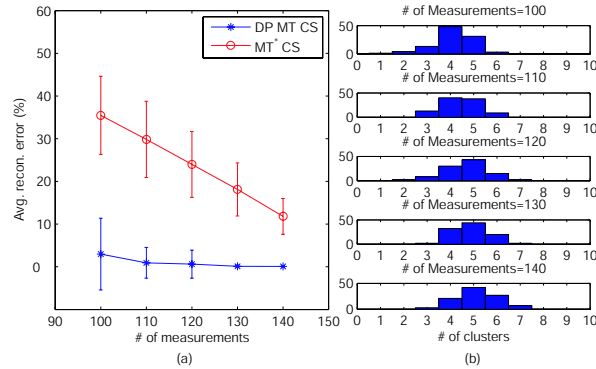


Fig. 4

MULTI-TASK CS INVERSION ERROR (IN PERCENT) FOR DP-BASED INVERSION AND WHEN A SHARED SPARSENESS PRIOR IS CONSIDERED. THESE RESULTS ARE FOR THE CASE IN WHICH THE DATA WERE GENERATED BASED ON FIVE CLUSTERS, WITH FIVE SPARSE SIGNALS GENERATED RANDOMLY FOR EACH OF THE FIVE TEMPLATES (SEE FIGURE 2). RESULTS ARE SHOWN FOR 100 DIFFERENT RANDOM GENERATIONS OF THE DATA. (A) RECONSTRUCTION ERRORS, (B) HISTOGRAMS OF THE NUMBER OF CLUSTERS YIELDED BY DP-MT.

prominent relative to single-task CS inversion when the number of CS measurements is diminished (as an aside, the improvement in the reconstruction error for DP-based CS relative to single-task CS is even more dramatic if the sparse signals have ± 1 values for non-zero components as in [23], as opposed to draws from $\mathcal{N}(0, 1)$).

To further examine the impact of the number of underlying clusters for data generation, we now consider examples for which the data are generated for 5, 3, 2 and 1 underlying clusters, with this performed as follows. Referring to Figure 2, for the 5-cluster case the top-left and top-right signals in Figure 2 are added, and then non-zero amplitudes from the sum are removed randomly, to generate a new template with 30 non-zero coefficients. This is done for each of the left-right pairs in Figure 2, to generate five template sparse signals. For each of these templates, five sparse signals are generated randomly, in the manner discussed above for the ten-cluster case; therefore, for the five-cluster case a total of $M = 25$ sparse signals are constituted, with which CS projection measurements of the type discussed above are performed. Similar merging of templates are used to constitute the 3, 2 and 1 cluster cases, again with five sparse signals generated randomly for each cluster. In Figures 4-7 are shown results in the form considered in Figure 3, for the case of 5, 3, 2 and 1 underlying clusters for data generation. One notes

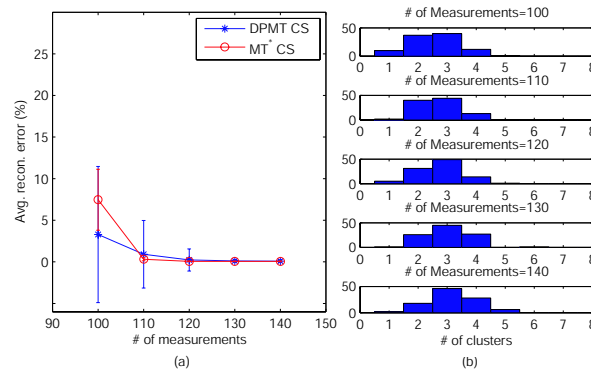


Fig. 5

MULTI-TASK CS INVERSION ERROR (IN PERCENT) FOR DP-BASED INVERSION AND WHEN A SHARED SPARSENESS PRIOR IS CONSIDERED. THESE RESULTS ARE FOR THE CASE IN WHICH THE DATA WERE GENERATED BASED ON THREE CLUSTERS, WITH FIVE SPARSE SIGNALS GENERATED RANDOMLY FOR EACH OF THE THREE TEMPLATES (SEE FIGURE 2). RESULTS ARE SHOWN FOR 100 DIFFERENT RANDOM GENERATIONS OF THE DATA. (A) RECONSTRUCTION ERRORS, (B) HISTOGRAMS OF THE NUMBER OF CLUSTERS YIELDED BY DP-MT.

the following phenomenon: As the number of underlying clusters diminishes, the difference between the DP-based and the global-sharing multi-task algorithms diminishes, with almost identical performance witnessed for the case of three and two clusters; this phenomenon is particularly evident as the number of CS measurements increases. As an aside, we also note that the DP-based inference of the number of underlying clusters adapts well to the underlying data generation.

We now provide an explanation for the relationships between the DP-based and the global-sparseness-sharing multi-task CS algorithms, as witnessed in Figures 3-6. For sparse signals like those in Figure 2 one's eyes are often drawn to the non-zero spikes in the signals; if these spikes between two signals are uncorrelated, as in the templates discussed above, the signals have distinct non-zero coefficients, and therefore one would typically infer that they have dissimilar sparseness properties. However, if two signals are very sparse, even when they have entirely dissimilar non-zero coefficients, they share many zero-amplitude coefficients. If we consider M sparse signals, and if *all* of the M signals share the same large set of zero-amplitude coefficients, then they are appropriate for sharing even if the associated (small number of) non-zero coefficients are entirely distinct. Returning to the ten-template case in Figure 2,

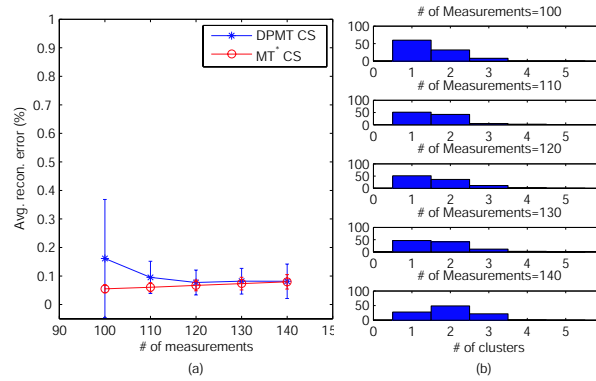


Fig. 6

MULTI-TASK CS INVERSION ERROR (IN PERCENT) FOR DP-BASED INVERSION AND WHEN A SHARED SPARSENESS PRIOR IS CONSIDERED. THESE RESULTS ARE FOR THE CASE IN WHICH THE DATA WERE GENERATED BASED ON TWO CLUSTERS, WITH FIVE SPARSE SIGNALS GENERATED RANDOMLY FOR EACH OF THE TWO TEMPLATES (SEE FIGURE 2). RESULTS ARE SHOWN FOR 100 DIFFERENT RANDOM GENERATIONS OF THE DATA. (A) RECONSTRUCTION ERRORS, (B) HISTOGRAMS OF THE NUMBER OF CLUSTERS YIELDED BY DP-MT.

because of the large number of clusters, the templates do not cumulatively share the same set of zero-amplitude coefficients; in this case global sharing for CS inversion is inappropriate, and the same is true for the five-cluster case considered in Figure 4. However, for the three and two cluster cases in Figures 5 and 6, since the number of distinct templates is small, and because there is substantial sparseness, the templates share a significant number of zero-amplitude coefficients, and therefore global sharing is appropriate (as in [23], [2], [39]). This underscores that global sharing across M tasks is appropriate when there is substantial sharing of zero-amplitude coefficients, even when all of the non-zero-amplitude coefficients are distinct (in effect, all of the tasks are used to jointly infer the shared zero-amplitude coefficients, and then the task-specific data are used to infer the relatively small set of unique non-zero coefficients).

To examine this sharing mechanism further, with an eye toward the real-image results presented below, we consider the sharing mechanisms manifested for two examples from the three-cluster case considered in Figure 5. The truncation level K can be set either to a large number or be estimated in principle by increase the number of sticks included until the log-marginal likelihood (the lower bound) in the VB algorithm starts to decrease. In this example we choose the number of sticks in the DP formulation to $K = 8$ which corresponds to the upper bound of the log-

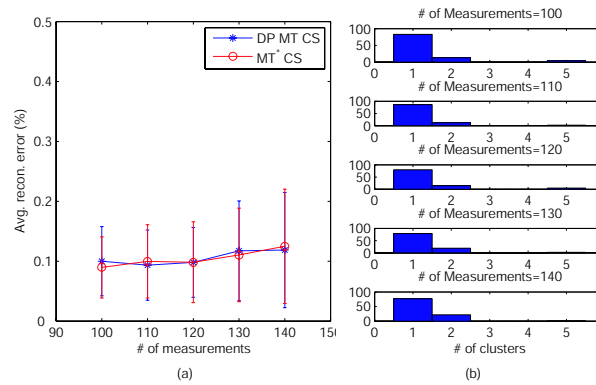


Fig. 7

MULTI-TASK CS INVERSION ERROR (IN PERCENT) FOR DP-BASED INVERSION AND WHEN A SHARED SPARSENESS PRIOR IS CONSIDERED. THESE RESULTS ARE FOR THE CASE IN WHICH THE DATA WERE GENERATED BASED ON ONE CLUSTER, WITH FIVE SPARSE SIGNALS GENERATED RANDOMLY FOR THE ASSOCIATED TEMPLATE. RESULTS ARE SHOWN FOR 100 DIFFERENT RANDOM GENERATIONS OF THE DATA. (A) RECONSTRUCTION ERRORS, (B) HISTOGRAMS OF THE NUMBER OF CLUSTERS YIELDED BY DP-MT.

marginal likelihood, and we show the stick (cluster) with which each of the 15 tasks were grouped at the end of the inference process. These examples were selected because they both yielded roughly the same average CS inversion accuracy across the 15 CS inversions (0.40% and 0.38% error), but we wish to emphasize that these two runs yield distinct clusterings. In both cases two clusters were manifested; however, for case (a) in Figure 8 tasks 1-5 and 11-15 are clustered together (although these were respectively generated by distinct and uncorrelated sparse templates), while in (b) for the most part tasks 1-10 are clustered together. This example is meant to emphasize that because the underlying signals are very sparse, and because they have significant overlap in the set of zero-amplitude coefficients, the particular clustering manifested by the DP formulation is not particularly important for the final CS-inversion quality. In fact, for these examples the global-sharing approaches in [23], [2], [39] would be effective, which as discussed above explains the good global-sharing performance in Figure 5. However, one typically does not know *a priori* if global sharing is appropriate (as it was *not* in Figures 3 and 4), and therefore the DP-based formulation offers generally high-quality results when global sharing is appropriate *and* when it is not.

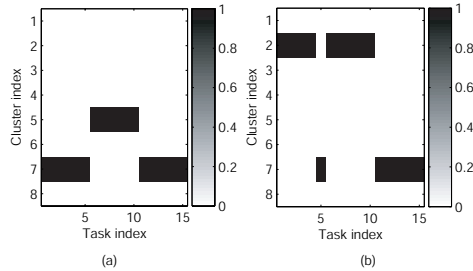


Fig. 8

TWO EXAMPLE RUNS OF THE DP-BASED MULTI-TASK CLUSTERING FOR THE 3-CLUSTER CASE OF DATA GENERATION, BASED ON 100 CS MEASUREMENTS. THE GREY SCALE DENOTES THE PROBABILITY THAT A GIVEN TASK IS ASSOCIATED WITH A PARTICULAR CLUSTER. (A) CASE FOR WHICH THE AVERAGE RECONSTRUCTION ERROR WAS 0.40%, (B) AVERAGE RECONSTRUCTION ERROR OF 0.38%.

B. Real images

The examples in Section V-A were selected as to provide an intuitive demonstration of multi-task CS, providing a comparison to DP-based and global sharing across tasks. Moreover, in this comparison DP-based inference was employed throughout, so that the only distinctions were the underlying model, and not the inference method. For practical application of MTL CS inversion, computational speed may be an issue, and this has motivated the simplified DP inference approximation discussed in Section IV. In the following examples, applied to imagery, we evaluate the performance of the two multi-task CS algorithms developed in this paper: DP-MT and the efficient SimDP-MT CS. We also perform comparisons with a multi-task CS framework that assumes complete sharing between the tasks (denoted as MT*, and discussed in Section II-A and considered in the examples above), and also a single-task Bayesian CS (ST), in which the CS inversion is performed independently on each of the tasks. Both MT* and ST are realized with the same algorithm as DP-MT (the VB DP-MT algorithm described in Section III) but with different settings, as discussed below.

We conduct two examples on CS reconstruction of typical imagery from “natural” scenes. All the images in these examples are of size 256×256 and are highly compressible in a wavelet basis. We choose the “Daubechies 8” wavelet as our orthonormal basis, and the sensing matrix Φ is constructed from a uniform spherical ensemble, which means each column of Φ (an $n \times 1$ vector) is uniformly distributed on the sphere S^{n-1} [12]. In this experiment we adopt a hybrid

CS scheme, in which using CS we measure only fine-scale wavelet coefficients, while retaining all coarse-scale coefficients (no compression in the coarse scale) [36]. In both examples, the coarsest scale is $j_0 = 3$, and the finest scale is $j_1 = 6$. For the DP-MT initialization, we set the hyper-parameters $a = 1/\text{std}(\mathbf{v})^2$ (std denotes the standard deviation), $b = 10^{-3}$, and $c = d = e = f = 10^{-4}$. The membership $\kappa_{i,k}$ for the task i is initialized randomly but with $\sum_{k=1}^K \kappa_{i,k} = 1$ and $\kappa_{i,k} \geq 0$. We use the mean of the posterior over $\boldsymbol{\theta}$ to perform the image reconstruction. For the SimDP-MT algorithm, $\nu = 1/\text{std}(\mathbf{v})^2$ and $\kappa_{i,k}$ is initialized by K-means clustering on the Euclidean distances between $\boldsymbol{\theta}$ s of any two original images estimated via the least square error (LSE) method ($\boldsymbol{\theta} = \Phi^+ \mathbf{v}$ where Φ^+ represents the pseudo inverse of Φ). For MT*, we employ the DP-MT algorithm, however we set $\kappa_{i,1} = 1$, and $\kappa_{i,k} = 0$ for $k > 1$ for all tasks; we also fix the values of $\kappa_{i,k}$ in each iteration without update. For ST, we still employ the DP-MT algorithm, but set $\kappa_{i,1} = 1$, and $\kappa_{i,k} = 0$ for $k > 1$, and consider only one CS task at a time ($M = 1$). The reconstruction error is defined as $\|\hat{\mathbf{u}} - \mathbf{u}\|_2 / \|\mathbf{u}\|_2$, where $\hat{\mathbf{u}}$ is the reconstructed image and \mathbf{u} is the original one.

In the first example, we choose 12 images from three different scenes. In the hybrid scheme, we assume all the wavelet coefficients at the finest scale are zero and only consider (estimate) the other 4096 coefficients. To reconstruct the image, we perform an inverse wavelet transform on the CS-estimated coefficients. In column (a) of Figure 9 we show the reconstructed images with all 4096 measurements using linear reconstruction ($\boldsymbol{\theta} = \Phi^T \mathbf{v}$), which is the best possible performance. Columns (b)-(e) in Figure 9 represent the reconstructed images by the DP-MT, SimDP-MT, MT*, and the ST algorithms, respectively, with the number of CS measurements $n = 1764$ (1700 measurements in the fine scales and 64 in the coarse scale) for each task. The reconstruction errors for these five methods are compared in Table II. We notice that both the DP-MT and SimDP-MT algorithms reduce the reconstruction error compared to the ST method, which indicates that the multi-task CS inversion shares information among tasks and therefore requires less measurements than the single task learning does to achieve the same performance. In addition to the CS inversion, the two new multi-task CS algorithms also yield task clustering, with this inferred simultaneously with the CS inversion; while this clustering is not the final product of interest, it is informative, with results shown in Figure 10. Note that the algorithms infer three clusters, each corresponding to a particular class of imagery. By contrast the MT* algorithm impose complete sharing among all tasks, and the results in Table I indicate that this

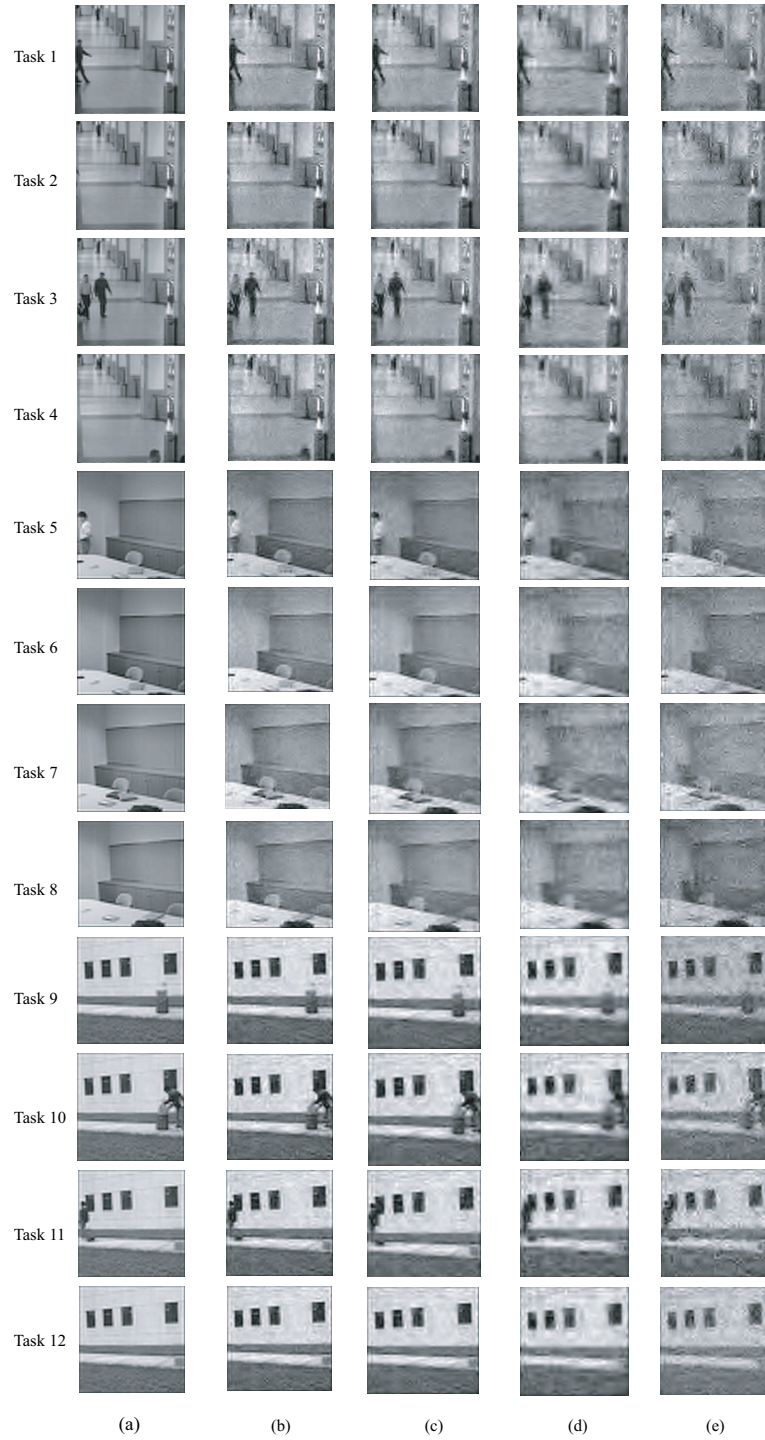


Fig. 9

CS RECONSTRUCTIONS, WHERE IN (A) 4096 MEASUREMENTS ARE EMPLOYED FOR EACH, WHILE IN (B)-(E) 1764 MEASUREMENTS ARE USED FOR EACH TASK. (A) LINEAR RECONSTRUCTION, (B) DP-MT, (C) SIMDP-MT, (D) MT*, (E) ST

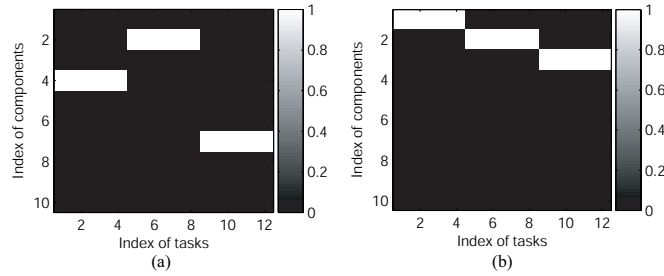


Fig. 10

SHARING MECHANISM FOR 12 TASKS FOR THE EXAMPLE IN FIGURE 9, WHERE ELEMENT (i, j) IN THE FIGURE REPRESENTS THE PROBABILITY OF TASK j BELONGS TO i -TH COMPONENT. (A) DP-MT, (B) SIMDP-MT.

undermines performance; DP-MT slightly outperforms SimDP-MT, since the former considers all possible values of the α_k^* during the estimation process while the latter yields only a point estimate of α_k^* . However, the SimDP-MT algorithm is much faster than DP-MT, especially when the data size is relatively large, as considered here. This example was run in MatlabTM on a Pentium IV PC with a 1.73 GHz CPU, where SimDP-MT requires roughly 4 hours while DP-MT needs about 16 hours (recall that we are simultaneously processing 12 images).

In the second example we consider 11 images from three scenes. The reconstructed images are shown in Figure 11 by the linear reconstruction, DP-MT, SimDP-MT, MT* and the ST algorithms; the reconstruction errors are listed in Table III for all five methods. As expected, the multi-task CS inversion algorithms yield smaller reconstruction error than the single task one,

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10	Task 11	Task 12
DP-MT	8.79	7.89	9.69	8.04	14.33	13.22	15.18	14.54	15.51	16.71	16.11	15.19
SimDP-MT	8.81	7.87	9.73	8.02	14.22	13.09	14.51	13.89	15.51	16.71	16.11	15.19
MT*	10.19	9.14	11.49	9.18	16.94	15.59	17.46	16.50	18.62	19.82	19.34	18.03
ST	10.28	10.37	12.81	10.28	18.37	16.18	18.65	17.67	20.77	22.24	21.19	19.59
Linear	6.66	6.20	7.08	6.14	12.41	11.70	12.43	11.99	13.83	14.41	14.10	13.53

TABLE II

RECONSTRUCTION ERROR (%) FOR THE EXAMPLE IN 9, IN WHICH DP-MT REPRESENTS THE VB DP-BASED MULTI-TASK CS ALGORITHM DISCUSSED IN SECTION III, SIMDP-MT REPRESENTS THE SIMPLIFIED DP MULTI-TASK CS ALGORITHM INTRODUCED IN SECTION IV, ST ADOPTS THE VB DP-MT CS ALGORITHM WHILE ASSUMING ONLY ONE TASK, MT* IS THE MULTI-TASK CS WITH GLOBAL SHARING AS DISCUSSED IN SECTION II-A, AND ADOPTS THE VB DP-MT CS ALGORITHM WHILE ASSUMING COMPLETELY SHARING, AND LINEAR IS DIRECT INVERSION TO ESTIMATE θ .

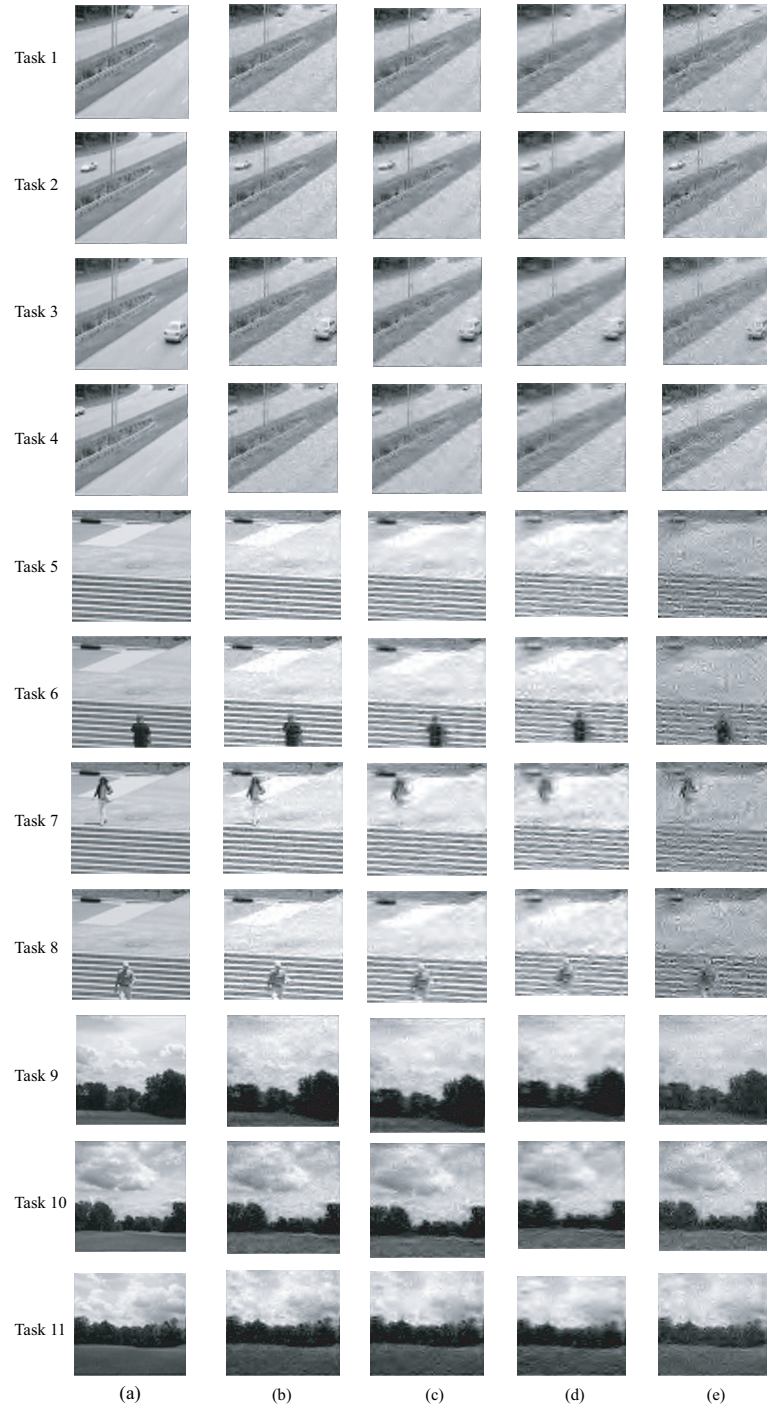


Fig. 11

CS RECONSTRUCTIONS, WHERE IN (A) 4096 MEASUREMENTS ARE EMPLOYED FOR EACH, WHILE IN (B)-(E) 1764 MEASUREMENTS ARE USED FOR EACH TASK. (A) LINEAR RECONSTRUCTION, (B) DP-MT, (C) SIMDP-MT, (D) MT*, (E) ST

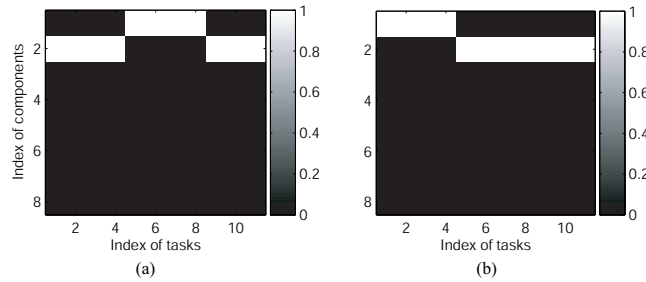


Fig. 12

SHARING MECHANISM FOR 11 TASKS IN FIGURE 11. (A) DP-MT, (B) SIMDP-MT.

and the relative performance among the different multi-task algorithms is consistent with Table II. Seemingly “confusing” clustering results are shown in Figure 12, in which images 1-4 and 9-11 are clustered together by DP-MT, while images 5-8 and 9-11 are clustered together by SimDP-MT. However, recall the simple example considered in Figure 8. The DP-based algorithm seeks to share the underlying sparseness of the images, even though the images themselves may appear distinct. The example in Figure 8 also underscored that there may be different types of MTL sharing manifested within the algorithm, but that the final CS inversions are often unaffected by this variation. In fact, the results in Figure 12 motivated the simple example considered in Figure 8.

VI. CONCLUSIONS

The problem of inverting multiple CS measurements has been considered from a Bayesian standpoint. Hierarchical priors are considered for the imposition of sparseness on the transform coefficients, with this performed in several ways. In the context of single-task (ST) learning, an

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6	Task 7	Task 8	Task 9	Task 10	Task 11
DP-MT	6.50	6.41	6.89	6.86	15.81	15.09	15.91	14.74	7.74	8.05	8.50
SimDP-MT	6.59	6.46	7.01	7.02	16.32	15.60	16.65	15.24	8.32	8.79	10.63
MT*	7.79	7.76	8.12	8.32	18.17	17.70	18.66	17.13	8.87	9.16	9.97
ST	8.31	8.23	8.81	9.23	19.79	19.74	20.36	18.88	8.77	9.58	9.62
Linear	4.78	4.77	5.01	5.15	15.39	14.49	15.18	14.06	6.10	5.72	6.72

TABLE III

RECONSTRUCTION ERROR (%) FOR THE EXAMPLE IN FIGURE 11, IN WHICH DP-MT, SIMDP-MT, ST, MT*, AND LINEAR

ARE DEFINED AS THE SAME AS IN TABLE I

independent zero-mean Gaussian prior is placed on each transform coefficient, and a Gamma prior is placed independently on the precision of each Gaussian. The parameters of these Gamma priors are set such that most precisions are likely to be very large, and therefore it is likely that most of the associated coefficients will be zero. This hierarchical prior yields a Student-t distribution on the transform coefficients. In the context of inverting multiple CS measurements, the aforementioned task-dependent precision parameters are assumed drawn from a distribution G , where G is drawn from a Dirichlet process (DP); the base distribution of the DP is of the same form as that discussed above for ST CS inference. The DP framework imposes the belief that many of the tasks may share underlying sparseness properties, and the objective is to cluster the CS measurements, where each cluster constitutes a particular form of sparseness. The DP formulation is non-parametric, in the sense that the number of clusters is not set *a priori* and is inferred from the data. The innovation parameter on the DP may be set as $\lambda = 0$, which reduces to all of the M CS inversions being inverted jointly (one global cluster), with this corresponding to the global sharing considered in previous multi-task CS inversions [23], [2], [39]. To make the Bayesian inference relatively computationally efficient, variational Bayesian inference has been considered on all model parameters, and we have also considered a simplified solution in which a point estimate is performed on the aforementioned precision parameters (with this termed a simplified DP solution).

For all examples considered, for both synthesized data and real imagery, the DP-based multi-task CS inversion performed at least as well as ST CS inversion. The utility of the DP-based inversion was most evident relative to ST inversion when the number of CS measurements was relatively small. The DP-based multi-task CS inversion always performed at least as well and generally better than CS inversion based on global sharing. When global sharing was appropriate, such a framework performed well; however, when global sharing was inappropriate, the DP-based inversion is significantly better.

An important observation from this study concerned when sharing of CS measurements is appropriate when performing CS inversion. Specifically, if two or more signals are very sparse, it is likely that CS-data sharing will be appropriate, even if none of the non-zero coefficients are the same. This is because the joint CS inversion is effective in inferring which set of (shared) coefficients are zero; after this information is inferred jointly from all of the data, effectively the data from individual CS measurements are used to infer the associated task-specific non-zero

coefficients and their amplitudes. Therefore, sharing of CS measurements may be more broadly appropriate than one may infer just based on looking at the data (*e.g.*, imagery) alone, particularly when there is a significant level of sparseness.

Concerning future research, as indicated above the effective Student-t distribution is imposed (within the prior) independently on each of the transform coefficients. When considering a wavelet transform of real signals or images, there is often significant correlation between spatially and spectrally adjacent wavelet coefficients. In future research it may be desirable to impose this knowledge directly within the prior. For example, rather than placing a Student-t distribution independently on the wavelet-transform coefficients, one may do something similar on the overall wavelet quad-tree. This will impose the concept that often entire quad-trees of a wavelet transform are sparse or not-sparse [9]. Similarly, it has been demonstrated that the hidden Markov tree (HMT) [9] provides a good statistical representation of wavelet coefficients. It is of interest to impose this statistical representation within the class of hierarchical Bayesian models considered in this paper. It is well known that the hidden Markov model, to which the HMT is related, is also in a form that is amenable to hierarchical Bayesian inference [32]. In future research we intend to integrate this class of hierarchical models [32] with the sharing mechanisms and sparseness promotion associated with the hierarchical models presented here.

In addition, within the DP-based inference algorithm employed here, it was assumed the “tasks” or CS measurements were exchangeable. This implies that the ordering of the different tasks is unimportant. In practical examples one may realize CS measurements sequentially in time, and/or the measurements may be constituted with a known spatial distribution. It is intuitive that the probability that two CS tasks are related should increase with their temporal or spatial proximity. To exploit such information, it is desirable to remove the assumption of exchangeability employed within the DP. Examples of how DP has been generalized for such purposes are discussed in (for example) [15], [19], [14]

REFERENCES

- [1] R. G. Baraniuk, “Compressive sensing,” *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, July 2007.
- [2] D. Baron, M. B. Wakin, M. F. Duarte, S. Sarvotham, and R. G. Baraniuk, “Distributed compressed sensing,” Nov. 2005.
- [3] M. J. Beal, “Variational algorithms for approximate bayesian inference,” Ph.D. dissertation, Gatsby Computational Neuroscience Unit, University College London, 2003.

- [4] D. Blackwell and J. MacQueen, "Ferguson distributions via polya urn schemes," *Annals of Statistics*, vol. 1, pp. 353–355, 1973.
- [5] D. Blei and M. Jordan, "Variational methods for the dirichlet process," *In Proceedings of the 21st International Conference on Machine Learning*, 2004.
- [6] E. Candes, "Compressive sensing," *Proceedings of the International Congress of Mathematics*, vol. 3, pp. 1433–1452, 2006.
- [7] E. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Inform. Theory*, vol. 52, pp. 489–502, 2006.
- [8] C. Charilaos, "Jpeg2000 tutorial," *ICIP*, 1999. [Online]. Available: www.dsp.toronto.edu/~dsp/JPEG2000
- [9] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Trans. on Signal Processing*, vol. 46, pp. 886–902, 1998.
- [10] I. Daubechies, "Ten lectures on wavelets," *SIAM journals*, 1992.
- [11] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of Royal Statistical Society B*, vol. 39, no. 1, pp. 1–38, 1977.
- [12] D. L. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [13] D. L. Donoho, Y. Tsaig, I. Drori, and J.-C. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," 2006, preprint.
- [14] D. Dunson, "Bayesian dynamic modeling of latent trait distributions," *Biostatistics*, 2006.
- [15] D. Dunson and J.-H. Park, "Kernel stick-breaking process," *Biometrika*, submitted.
- [16] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, vol. 90, no. 430, pp. 577–588, 1995.
- [17] T. S. Ferguson, "A bayesian analysis of some nonparametric problems," *Annals of Statistics*, vol. 1, pp. 209–230, 1973.
- [18] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, "Introducing markov chain monte carlo," in *Markov Chain Monte Carlo in Practice*. London, U.K.: Chapman Hall, 1996.
- [19] J. Griffin and M. Steel, "Order-based dependent Dirichlet process," *Journal of the American Statistical Association*, p. in press, 2006.
- [20] T. L. Griffiths and M. Steyvers, "Finding scientific topics," in *PNAS*, 2004.
- [21] H. Ishwaran and L. F. James, "Gibbs sampling methods for stick-breaking priors," *Journal of the American Statistical Association, Theory and Methods*, vol. 96, no. 453, pp. 161–173, 2001.
- [22] H. Ishwaran and Zarepour, "Exact and approximate sum-representations for the dirichlet process," *Canadian Journal Of Statistics*, vol. 30, pp. 269–283, 2002.
- [23] S. Ji, D. Dunson, and L. Carin, "Multi-task compressive sensing," *Submit to IEEE Trans. on Signal Processing*, 2007.
- [24] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *Accepted to IEEE Trans. on Signal Processing*, 2007.
- [25] D. J. C. MacKay, "Bayesian methods for backpropagation networks," in *Models of Neural Networks III*, E. Domany, J. L. van Hemmen, and K. Schulten, Eds. New York: Springer-Verlag, 1994.
- [26] R. M. Neal, *Bayesian Learning for Neural Networks*. Springer, 1996.
- [27] 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 for Video Technology*, vol. 14, pp. 1219–1235, 2004.
- [28] D. L. D. S. Chen and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1999.

- [29] 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.
- [30] J. Sethuraman, "A constructive definition of the dirichlet prior," *Statistica Sinica*, vol. 2, pp. 639–650, 1994.
- [31] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. University of Illinois, 1963.
- [32] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, pp. 1566–1581, 2006.
- [33] M. Tipping and A. Faul, "Fast marginal likelihood maximisation for sparse bayesian models," *Proc. of the 9-th Intern. Workshop on AI and Statistics*, 2003.
- [34] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, pp. 211–244, 2001.
- [35] J. A. Tropp and A. C. Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," 2005, preprint.
- [36] Y. Tsaig and D. L. Donoho, "Extensions of compressed sensing," *Signal Processing*, vol. 86, no. 3, pp. 549–571, March 2006.
- [37] G. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, April 1991.
- [38] M. West, P. Muller, and M. Escobar, "Hierarchical priors and mixture models with applications in regression and density estimation," in *Aspects of Uncertainty*, P. R. Freeman and A. F. Smith, Eds. John Wiley, 1994, pp. 363–386.
- [39] D. P. Wipf and B. D. Rao, "An empirical bayesian strategy for solving the simultaneous sparse approximation problem," *IEEE Trans. on Signal Processing*, vol. 55, no. 7, pp. 3704–3716, July 2007.

APPENDIX I

UPDATE EQUATIONS IN VB DP-MT CS

To update each $q(\cdot)$, only those terms related to $q(\cdot)$ in (21) are kept for the derivation. For example, to derive update equation for $q(\alpha_0)$ we have

$$\mathcal{F}(q(\alpha_0)) = \int_{\alpha_0} q(\alpha_0) \cdot \left[\log p(\alpha_0|a, b) + \sum_{i=1}^M \int_{\boldsymbol{\theta}_i} q(\boldsymbol{\theta}_i) \log p(\mathbf{v}_i|\boldsymbol{\theta}_i, \alpha_0) d\boldsymbol{\theta}_i - \log q(\alpha_0) \right] d\alpha_0 \quad (1)$$

We take the functional derivative with respect to $q(\alpha_0)$ and set it to zero, *i.e.*, $\partial\mathcal{F}(q(\alpha_0))/\partial q(\alpha_0) = 0$, to obtain the update equation for $q(\alpha_0)$. In a similar manner, the update equations for all $q(\cdot)$ are given as follows:

- $q(\alpha_0) \sim Ga(\tilde{a}, \tilde{b})$

$$\begin{aligned} \tilde{a} &= a + \frac{1}{2} \sum_{i=1}^M n_i \\ \tilde{b} &= b + \frac{1}{2} \sum_{i=1}^M [tr(\boldsymbol{\Phi}_i \boldsymbol{\Gamma}_i^{-1} \boldsymbol{\Phi}_i^T) + (\boldsymbol{\Phi}_i \boldsymbol{\mu}_i - \mathbf{v}_i)^T (\boldsymbol{\Phi}_i \boldsymbol{\mu}_i - \mathbf{v}_i)] \end{aligned} \quad (2)$$

- $q(\lambda) \sim Ga(\tilde{e}, \tilde{f})$

$$\begin{aligned} \tilde{e} &= e + K - 1 \\ \tilde{f} &= f - \sum_{k=1}^{K-1} [\psi(\tau_{2k}) - \psi(\tau_{1k} + \tau_{2k})], \end{aligned} \quad (3)$$

where $\psi(x) = \frac{\partial}{\partial x} \log \Gamma(x)$.

- $q(\pi_k) \sim Beta(\tau_{1k}, \tau_{2k}), \quad k = 1, \dots, K - 1$

$$\begin{aligned} \tau_{1k} &= 1 + \sum_{i=1}^M \kappa_{i,k} \\ \tau_{2k} &= \frac{\tilde{e}}{\tilde{f}} + \sum_{i=1}^M \sum_{l=k+1}^K \kappa_{i,l}, \end{aligned} \quad (4)$$

where $\kappa_{i,k} = q(z_i = k)$.

- $q(\alpha_{k,j}^*) \sim Ga(\tilde{c}_{k,j}, \tilde{d}_{k,j}), \quad j = 1, \dots, m, \quad k = 1, \dots, K$

$$\begin{aligned} \tilde{c}_{k,j} &= c + \frac{1}{2} \sum_{i=1}^M \kappa_{i,k} \\ \tilde{d}_{k,j} &= d + \frac{1}{2} \sum_{i=1}^M \kappa_{i,k} (\sigma_{i,j} + \mu_{i,k,j}^2), \end{aligned} \quad (5)$$

where $[\sigma_{i,1}, \dots, \sigma_{i,m}]$ is the diagonal elements of $\mathbf{\Gamma}_i^{-1}$ and $\mu_{i,k,j}$ is the j^{th} element of vector $\boldsymbol{\mu}_i$.

- $q(\boldsymbol{\theta}_i) \sim \mathcal{N}(\boldsymbol{\mu}_i, \mathbf{\Gamma}_i^{-1}), \quad i = 1, \dots, M$

$$\begin{aligned}\mathbf{\Gamma}_i &= \sum_{k=1}^K \kappa_{i,k} \mathbf{\Lambda}_k + \frac{\tilde{a}}{\tilde{b}} \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_i \\ \boldsymbol{\mu}_i &= \frac{\tilde{a}}{\tilde{b}} \mathbf{\Gamma}_i^{-1} \boldsymbol{\Phi}_i^T \mathbf{v}_i,\end{aligned}\quad (6)$$

where $\mathbf{\Lambda}_k = \text{diag}(\tilde{c}_{k,1}/\tilde{d}_{k,1}, \dots, \tilde{c}_{k,m}/\tilde{d}_{k,m})$ is a diagonal matrix of $m \times m$.

- $q(z_i = k) = \kappa_{i,k}, \quad i = 1, \dots, M, \quad k = 1, \dots, K$

$$\kappa_{i,k} = \frac{e^{\lambda_{i,k}}}{\sum_{l=1}^K e^{\lambda_{i,l}}}, \quad (7)$$

where

$$\begin{aligned}\lambda_{i,k} &= \sum_{l=1}^{k-1} [\psi(\tau_{2l}) - \psi(\tau_{1l} + \tau_{2l})] + [\psi(\tau_{1k}) - \psi(\tau_{1k} + \tau_{2k})] - \\ &\quad - \frac{1}{2} \left\{ \sum_{j=1}^m [\ln 2\pi - \psi(\tilde{c}_{k,j}) + \ln(\tilde{d}_{k,j})] + \text{tr}(\mathbf{\Gamma}_i^{-1} \mathbf{\Lambda}_k) + \boldsymbol{\mu}_i^T \mathbf{\Lambda}_k \boldsymbol{\mu}_i \right\}\end{aligned}\quad (8)$$

The lower bound $\mathcal{F}(q)$ can be rearranged in the following form

$$\mathcal{F} = \sum_{i=1}^{10} \mathcal{F}_i + \text{const}, \quad (9)$$

where

$$\mathcal{F}_1 = -KL(q(\alpha_0) \| p(\alpha_0 | a, b)) = -KL_{Gamma}(\tilde{a}, \tilde{b} \| a, b), \quad (10)$$

$$\mathcal{F}_2 = -KL_{Gamma}(\tilde{d}, \tilde{f} \| c, d), \quad (11)$$

$$\mathcal{F}_3 = - \sum_{k=1}^K \sum_{j=1}^m KL_{Gamma}(\tilde{c}_{k,j}, \tilde{d}_{k,j} \| c, d), \quad (12)$$

$$\begin{aligned}\mathcal{F}_4 &= \int \int q(\mathbf{w}) q(\lambda) \ln p(\mathbf{w} | \lambda) d\mathbf{w} d\lambda, \\ &= (K-1)[\psi(\tilde{e}) - \psi(\tilde{f})] + \left(\frac{\tilde{e}}{\tilde{f}} - 1\right) \sum_{k=1}^{K-1} [\psi(\tau_{2k}) - \psi(\tau_{1k} + \tau_{2k})],\end{aligned}\quad (13)$$

$$\begin{aligned}
\mathcal{F}_5 &= \sum_{i=1}^M \int \int q(z_i) q(\mathbf{w}) \ln p(z_i | \mathbf{w}) dz_i d\mathbf{w}, \\
&= \sum_{i=1}^M \sum_{k=1}^K \kappa_{i,k} \left\{ \sum_{l=1}^{k-1} [\psi(\tau_{2l}) - \psi(\tau_{1l} + \tau_{2l})] + \psi(\tau_{1k}) - \psi(\tau_{1k} + \tau_{2k}) \right\}, \quad (14)
\end{aligned}$$

$$\begin{aligned}
\mathcal{F}_6 &= \sum_{i=1}^M \sum_{k=1}^K \int \int \int q(\boldsymbol{\theta}_i) q(z_i) q(\boldsymbol{\alpha}_k^*) \ln(p(\boldsymbol{\theta}_i | z_i, \boldsymbol{\alpha}_k^*)) d\boldsymbol{\theta}_i dz_i d\boldsymbol{\alpha}_k^* \\
&= -\frac{1}{2} \sum_{i=1}^M \sum_{k=1}^K \kappa_{i,k} \left\{ \sum_{j=1}^m [\ln 2\pi - \psi(\tilde{c}_{k,j}) + \ln(\tilde{d}_{k,j})] + \text{tr}(\boldsymbol{\Gamma}_i^{-1} \boldsymbol{\Lambda}_k) + \boldsymbol{\mu}_i^T \boldsymbol{\Lambda}_k \boldsymbol{\mu}_i \right\}, \quad (15)
\end{aligned}$$

$$\begin{aligned}
\mathcal{F}_7 &= \sum_{i=1}^M \sum_{k=1}^K \int \int q(\boldsymbol{\theta}_i) q(\alpha_0) \ln(p(\mathbf{v}_i | \boldsymbol{\theta}_i, \alpha_0)) d\boldsymbol{\theta}_i d\alpha_0 \\
&= \frac{[\psi(\tilde{a}) - \ln \tilde{b} - \ln 2\pi]}{2} \sum_{i=1}^M n_i - \frac{\tilde{a}}{2\tilde{b}} \sum_{i=1}^M [\text{tr}(\boldsymbol{\Phi}_i \boldsymbol{\Gamma}_i^{-1} \boldsymbol{\Phi}_i^T) + (\boldsymbol{\Phi}_i \boldsymbol{\mu}_i - \mathbf{v}_i)^T (\boldsymbol{\Phi}_i \boldsymbol{\mu}_i - \mathbf{v}_i)] \quad (16)
\end{aligned}$$

$$\mathcal{F}_8 = - \int q(\mathbf{w}) \ln q(\mathbf{w}) d\mathbf{w} = - \sum_{k=1}^{K-1} [\psi(\tau_{1k}) - \psi(\tau_{1k} + \tau_{2k})], \quad (17)$$

$$\mathcal{F}_9 = - \sum_{i=1}^M \int q(z_i) \ln q(z_i) dz_i = - \sum_{i=1}^M \sum_{k=1}^K \kappa_{i,k} \ln(\kappa_{i,k}), \quad (18)$$

$$\mathcal{F}_{10} = - \sum_{i=1}^M \int q(\boldsymbol{\theta}_i) \ln q(\boldsymbol{\theta}_i) d\boldsymbol{\theta}_i = - \sum_{i=1}^M \ln |\boldsymbol{\Gamma}_i|. \quad (19)$$

APPENDIX II

UPDATE EQUATIONS IN THE SIMDP-MT CS ALGORITHM

To develop an efficient inference algorithm for the SimDP-MT CS, we introduce new quantities

$$S_{i,k,j} \triangleq \boldsymbol{\Phi}_{i,j}^T \mathbf{B}_{i,k}^{-1} \boldsymbol{\Phi}_{i,j}, \quad Q_{i,k,j} \triangleq \boldsymbol{\Phi}_{i,j}^T \mathbf{B}_{i,k}^{-1} \mathbf{v}_i, \quad \text{and} \quad G_{i,k} \triangleq \mathbf{v}_i^T \mathbf{B}_{i,k}^{-1} \mathbf{v}_i + \nu, \quad (20)$$

which can be updated sequentially, and therefore we have

$$s_{i,k,j} = \frac{\alpha_{k,j}^* S_{i,k,j}}{\alpha_{k,j}^* - S_{i,k,j}}, \quad q_{i,k,j} = \frac{\alpha_{k,j}^* Q_{i,k,j}}{\alpha_{k,j}^* - S_{i,k,j}}, \quad \text{and} \quad g_{i,k,j} = G_{i,k} + \frac{Q_{i,k,j}^2}{\alpha_{k,j}^* - S_{i,k,j}}. \quad (21)$$

The integer $k \in \{1, \dots, J\}$ is used to index the k^{th} mixture component, $j \in \{1, \dots, m\}$ to index the single basis function for which $\alpha_{k,j}^*$ is to be updated, t to index the current presented basis, and $l \in \{1, \dots, m\}$ to index all basis functions. Define $K_i = n_i + 2\nu$ for convenience.

A. Membership updating

$$\kappa_{i,k} = \frac{e^{\lambda_{i,k}}}{\sum_{l=1}^J e^{\lambda_{i,l}}}, \quad (22)$$

where

$$\lambda_{i,k} = \left[\psi(\omega_k) - \psi\left(\sum_{l=1}^J \omega_l\right) \right] - \frac{1}{2} \left[(n_i + \nu) \log(\mathbf{v}_i^T \mathbf{B}_{i,k}^{-1} \mathbf{v}_i + \frac{\nu}{2}) + \log |\mathbf{B}_{i,k}| \right]. \quad (23)$$

$$\omega_k = \frac{1}{J} + \sum_{i=1}^M \kappa_{i,k}, \quad \text{for } k = 1, \dots, J.$$

B. Adding a basis function (j^{th} basis) for k^{th} mixture component

$$\begin{aligned} 2\Delta\mathcal{L}_{i,k} &= \kappa_{i,k} \left[\log \frac{\alpha_{k,j}^*}{\alpha_{k,j}^* + s_{i,k,j}} - K_i \log\left(1 - \frac{q_{i,k,j}^2/g_{i,k,j}}{\alpha_{k,j}^* + s_{i,k,j}}\right) \right], \\ \tilde{\Sigma}_{i,k} &= \begin{bmatrix} \Sigma_{i,k} + \Sigma_{i,k,(jj)} \Sigma_{i,k} \Phi_i^T \Phi_{i,j} \Phi_i \Sigma_{i,k} & -\Sigma_{i,k,(jj)} \Sigma_{i,k} \Phi_i^T \Phi_{i,j} \\ -\Sigma_{i,k,(jj)} (\Sigma_{i,k} \Phi_i^T \Phi_{i,j})^T & \Sigma_{i,k,(jj)} \end{bmatrix}, \\ \tilde{\boldsymbol{\mu}}_{i,k} &= \begin{bmatrix} \boldsymbol{\mu}_{i,k} - \mu_{i,k,j} \Sigma_{i,k} \Phi_i^T \Phi_{i,j} \\ \mu_{i,k,j} \end{bmatrix}, \\ \tilde{S}_{i,k,l} &= S_{i,k,l} - \Sigma_{i,k,(jj)} (\Phi_{i,l}^T \mathbf{e}_{i,k,j})^2, \\ \tilde{Q}_{i,k,l} &= Q_{i,k,l} - \mu_{i,k,j} (\Phi_{i,l}^T \mathbf{e}_{i,k,j}), \\ \tilde{G}_{i,k} &= G_{i,k} - \Sigma_{i,k,(jj)} (\mathbf{v}_i^T \mathbf{e}_{i,k,j})^2, \end{aligned} \quad (24)$$

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

C. Re-estimating a basis function (t^{th} basis) for k^{th} mixture component

Define $a_{i,k,t} \triangleq (\Sigma_{i,k,(tt)} + (\tilde{\alpha}_{k,j}^* - \alpha_{k,j}^*)^{-1})^{-1}$ and $\Sigma_{i,k,t}$ as the k^{th} column of $\Sigma_{i,k}$:

$$\begin{aligned}
2\Delta\mathcal{L}_{i,k} &= \kappa_{i,k} \left[(K_i - 1) \log(1 + S_{i,j}(\tilde{\alpha}_{k,j}^{*-1} - \alpha_{k,j}^{*-1})) + K_i \log \frac{[(\alpha_{k,j}^* + s_{i,k,j})g_{i,k,j} - q_{i,k,j}^2] \tilde{\alpha}_{k,j}^*}{[(\tilde{\alpha}_{k,j}^* + s_{i,k,j})g_{i,k,j} - q_{i,k,j}^2] \alpha_{k,j}^*} \right], \\
\tilde{\Sigma}_{i,k} &= \Sigma_{i,k} - a_{i,k,t} \Sigma_{i,k,t} \Sigma_{i,k,t}^T, \\
\tilde{\boldsymbol{\mu}}_{i,k} &= \boldsymbol{\mu}_{i,k} - a_{i,k,t} \mu_{i,k,t} \Sigma_{i,k,t}, \\
\tilde{S}_{i,k,l} &= S_{i,k,l} + a_{i,k,t} (\Sigma_{i,k,t}^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_{i,l})^2, \\
\tilde{Q}_{i,k,l} &= Q_{i,k,l} + a_{i,k,t} \mu_{i,k,t} (\Sigma_{i,k,t}^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_{i,l}), \\
\tilde{G}_{i,k} &= G_{i,k} + a_{i,k,t} (\Sigma_{i,k,t}^T \boldsymbol{\Phi}_i^T \mathbf{v}_i)^2.
\end{aligned} \tag{25}$$

D. Deleting a basis function (t^{th} basis) for k^{th} mixture component

$$\begin{aligned}
2\Delta\mathcal{L}_{i,k} &= \kappa_{i,k} \left[-K_i \log\left(1 + \frac{Q_{i,k,j}^2/G_{i,k}}{\alpha_{k,j}^* - S_{i,k,j}}\right) - \log\left(1 - \frac{S_{i,k,j}}{\alpha_{k,j}^*}\right) \right], \\
\tilde{\Sigma}_{i,k} &= \Sigma_{i,k} - \frac{1}{\Sigma_{i,k,(tt)}} \Sigma_{i,k,t} \Sigma_{i,k,t}^T, \\
\tilde{\boldsymbol{\mu}}_{i,k} &= \boldsymbol{\mu}_{i,k} - \frac{\mu_{i,k,t}}{\Sigma_{i,k,(tt)}} \Sigma_{i,k,t}, \\
\tilde{S}_{i,k,l} &= S_{i,k,l} + \frac{1}{\Sigma_{i,k,(tt)}} (\Sigma_{i,k,t}^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_{i,l})^2, \\
\tilde{Q}_{i,k,l} &= Q_{i,k,l} + \frac{\mu_{i,k,t}}{\Sigma_{i,k,(tt)}} (\Sigma_{i,k,t}^T \boldsymbol{\Phi}_i^T \boldsymbol{\Phi}_{i,l}), \\
\tilde{G}_{i,k} &= G_{i,k} + \frac{1}{\Sigma_{i,k,(tt)}} (\Sigma_{i,k,t}^T \boldsymbol{\Phi}_i^T \mathbf{v}_i)^2.
\end{aligned} \tag{26}$$

After updating $\tilde{\Sigma}_{i,k}$ and $\tilde{\boldsymbol{\mu}}_{i,k}$, remove the corresponding row and/or column t from $\tilde{\Sigma}_{i,k}$ and $\tilde{\boldsymbol{\mu}}_{i,k}$.