Bayesian Nonparametric Dictionary Learning for Compressed Sensing MRI

Yue Huang, John Paisley, Qin Lin, Xinghao Ding, Xueyang Fu, and Xiao-Ping Zhang, Senior Member, IEEE

Abstract—We develop a Bayesian nonparametric model for reconstructing magnetic resonance images (MRIs) from highly undersampled k-space data. We perform dictionary learning as part of the image reconstruction process. To this end, we use the beta process as a nonparametric dictionary learning prior for representing an image patch as a sparse combination of dictionary elements. The size of the dictionary and patch-specific sparsity pattern are inferred from the data, in addition to other dictionary learning variables. Dictionary learning is performed directly on the compressed image, and so is tailored to the MRI being considered. In addition, we investigate a total variation penalty term in combination with the dictionary learning model, and show how the denoising property of dictionary learning removes dependence on regularization parameters in the noisy setting. We derive a stochastic optimization algorithm based on Markov chain Monte Carlo for the Bayesian model, and use the alternating direction method of multipliers for efficiently performing total variation minimization. We present empirical results on several MRI, which show that the proposed regularization framework can improve reconstruction accuracy over other methods.

Index Terms—Compressed sensing, magnetic resonance imaging, Bayesian nonparametrics, dictionary learning.

I. INTRODUCTION

MAGNETIC resonance imaging (MRI) is a widely used technique for visualizing the structure and functioning of the body. A limitation of MRI is its slow scan speed during data acquisition. Therefore, methods for accelerating the MRI process have been heavily researched. Recent advances in signal reconstruction from measurements sampled below the Nyquist rate, called compressed sensing (CS) [1], [2], have had a major impact on MRI [3]. CS-MRI allows for significant undersampling in the Fourier measurement domain of MR images (called k-space), while still outputting a high-quality image reconstruction. While image reconstruction using this undersampled data is a case of an ill-posed inverse problem, compressed sensing theory has shown that it is possible to reconstruct a signal from significantly fewer measurements than mandated by traditional Nyquist sampling if the signal is sparse in a particular transform domain.

Motivated by the need to find a sparse domain for representing the MR signal, a large body of literature now exists on reconstructing MRI from significantly undersampled k-space data. Existing improvements in CS-MRI mostly focus on (i) seeking sparse domains for the image, such as contourlets [4], [5]; (ii) using approximations of the ℓ₀ norm for better reconstruction performance with fewer measurements, for example ℓ₁, FOCUSS, ℓ_p quasi-norms with 0 < p < 1, or using smooth functions to approximate the ℓ₀ norm [6], [7]; and (iii) accelerating image reconstruction through more efficient optimization techniques [8], [10], [29].

In this paper we present a modeling framework that is similarly motivated.

CS-MRI reconstruction algorithms tend to fall into two categories: Those which enforce sparsity directly within some image transform domain [3]–[8], [10]–[12], and those which enforce sparsity in some underlying latent representation of the image, such as an adaptive dictionary-based representation [9], [14]. Most CS-MRI reconstruction algorithms belong to the first category. For example Sparse MRI [3], the leading study in CS-MRI, performs MR image reconstruction by enforcing sparsity in both the wavelet domain and the total variation (TV) of the reconstructed image. Algorithms with image-level sparsity constraints such as Sparse MRI typically employ an off-the-shelf basis, which can usually capture only one feature of the image. For example, wavelets recover point-like features, while contourlets recover curve-like features. Since MR images contain a variety of underlying features, such as edges and textures, using a basis not adapted to the image can be considered a drawback of these algorithms.

Finding a sparse basis that is suited to the image at hand can benefit MR image reconstruction, since CS theory shows that the required number of measurements is linked to the sparsity of the signal in the selected transform domain. Using a standard basis not adapted to the image under consideration will likely not provide a representation that can compete in sparsity with an adapted basis. To this end, dictionary learning,..
which falls in the second group of methods, learns a sparse basis on image subregions called patches that is adapted to the image class of interest. Recent studies in the image processing literature have shown that dictionary learning is an effective means for finding a sparse, patch-level representation of an image [19], [20], [25]. These algorithms learn a patch-level dictionary by exploiting structural similarities between patches extracted from images within a class of interest. Among these approaches, adaptive dictionary learning—where the dictionary is learned directly from the image being considered—based on patch-level sparsity constraints usually outperforms analytical dictionary approaches in denoising, super-resolution reconstruction, interpolation, inpainting, classification and other applications, since the adaptively learned dictionary suits the signal of interest [19]–[22].

Dictionary learning has previously been applied to CS-MRI to learn a sparse basis for reconstruction, see [14]. With these methods, parameters such as the dictionary size and patch sparsity are preset, and algorithms are considered that are non-Bayesian. In this paper, we consider a new dictionary learning algorithm for CS-MRI that is motivated by Bayesian nonparametric statistics. Specifically, we consider a nonparametric dictionary learning model called BPFA [23] that uses the beta process to learn the sparse representation necessary for CS-MRI reconstruction. The beta process is an effective prior for nonparametric learning of latent factor models; in this case the latent factors correspond to dictionary elements. While the dictionary size is therefore infinite in principle, through posterior inference the beta process learns a suitably compact dictionary in which the signal can be sparsely represented.

We organize the paper as follows. In Section II we review CS-MRI inversion methods and the beta process for dictionary learning. In Section III, we describe the proposed regularization framework and algorithm. We derive a Markov Chain Monte Carlo (MCMC) sampling algorithm for stochastic optimization of the dictionary variables in the objective function. In addition, we consider including a sparse total variation (TV) penalty, for which we perform efficient optimization using the alternating direction method of multipliers (ADMM). We then show the advantages of the proposed Bayesian nonparametric regularization framework on several CS-MRI problems in Section IV.

II. BACKGROUND AND RELATED WORK

We use the following notation: Let \(x \in \mathbb{C}^N\) be a \(\sqrt{N} \times \sqrt{N}\) MR image in vectorized form. Let \(\mathcal{F}_u \in \mathbb{C}^{u \times N}\), \(u \ll N\), be the undersampled Fourier encoding matrix and \(y = \mathcal{F}_u x \in \mathbb{C}^u\) represent the sub-sampled set of \(k\)-space measurements. The goal is to estimate \(x\) from the small fraction of \(k\)-space measurements \(y\). For dictionary learning, let \(R_i\) be the \(i^{th}\) patch extraction matrix. That is, \(R_i\) is a \(P \times N\) matrix of all zeros except for one in each row that extracts a vectorized \(\sqrt{P} \times \sqrt{P}\) patch from the image, \(R_i x \in \mathbb{C}^P\) for \(i = 1, \ldots, N\). We use overlapping image patches with a shift of one pixel and allow a patch to wrap around the image at the boundaries for mathematical convenience [15], [22]. All norms are extended to complex vectors when necessary, \(\|a\|_p = (\sum |a_i|^p)^{1/p}\), where \(|a_i|\) is the modulus of the complex number \(a_i\).

A. Two Approaches to CS-MRI Inversion

We focus on single-channel CS-MRI inversion via optimizing an unconstrained function of the form

\[
\arg\min_x h(x) + \frac{\lambda}{2} \|\mathcal{F}_u x - y\|_2^2,
\]

where \(\|\mathcal{F}_u x - y\|_2^2\) is a data fidelity term, \(\lambda > 0\) is a parameter and \(h(x)\) is a regularization function that controls properties of the image we want to reconstruct. As discussed in the introduction, the function \(h\) can take several forms, but tends to fall into one of two categories according to whether image-level or patch-level information is considered. We next review these two approaches.

1) Image-Level Sparse Regularization: CS-MRI with an image-level, or global regularization function \(h_g(x)\) is one in which sparsity is enforced within a transform domain defined on the entire image. For example, in Sparse MRI [3] the regularization function is

\[
h_g(x) = \|Wx\|_1 + \mu TV(x),
\]

where \(W\) is the wavelet basis and \(TV(x)\) is the total variation (spatial finite differences) of the image. Regularizing with this function requires that the image be sparse in the wavelet domain, as measured by the \(\ell_1\) norm of the wavelet coefficients \(\|Wx\|_1\), which acts as a surrogate for \(\ell_0\) [1], [2]. The total variation term enforces homogeneity within the image by encouraging neighboring pixels to have similar values while allowing for sudden high frequency jumps at edges. The parameter \(\mu > 0\) controls the trade-off between the two terms. A variety of other image-level regularization approaches have been proposed along these lines, see [4], [5], [7].

2) Patch-Level Sparse Regularization: An alternative to the image-level sparsity constraint \(h_g(x)\) is a patch-level, or local regularization function \(h_l(x)\), which enforces that patches (square sub-regions of the image) have a sparse representation according to a dictionary. One possible general form of such a regularization function is

\[
h_l(x) = \sum_{i=1}^N \gamma_i \|R_i x - D a_i\|_2^2 + f(a_i, D),
\]

where the dictionary matrix is \(D \in \mathbb{C}^{P \times K}\) and \(a_i\) is a \(K\)-dimensional vector in \(\mathbb{R}^K\). An important difference between \(h_l(x)\) and \(h_g(x)\) is the additional function \(f(a_i, D)\). While image-level sparsity constraints fall within a predefined transform domain, such as the wavelet basis, the sparse dictionary domain can be unknown for patch-level regularization and learned from data. The function \(f\) enforces sparsity by learning a \(D\) for which \(a_i\) is sparse.\(^1\) For example, [9] uses K-SVD to learn \(D\) off-line, and then approximately optimize the objective function

\[
\arg\min_{a_1, \ldots, a_N} \sum_{i=1}^N \|R_i x - D a_i\|_2^2 \text{ subject to } \|a_i\|_0 \leq T, \forall i,
\]

using orthogonal matching pursuits (OMP) [21]. In this case, the \(L_0\) penalty on the additional parameters \(a_i\) make this a

\(^1\)The dependence of \(h_l(x)\) on \(D\) and \(a_i\) is implied in our notation.
Algorithm 1 Dictionary Learning With BPFA

1) Construct a dictionary $D = [d_1, \ldots, d_K]$, with $d_k \sim N(0, P^{-1} I)$, $k = 1, \ldots, K$.

2) Draw a probability $\pi_k \in [0, 1]$ for each $d_k$:
$$\pi_k \sim \text{Beta}(c\gamma/K, c(1 - \gamma/K)), \quad k = 1, \ldots, K.$$

3) Draw precision values for noise and each weight $\gamma_i \sim \text{Gam}(90, 10)$, $\gamma_s \sim \text{Gam}(0.5, 0.5)$.

4) For the $i$th patch in $x$:
   a) Draw the vector $s_i \sim N(0, \gamma_s^{-1} I_K)$.
   b) Draw the binary vector $z_i$ with $z_{ik} \sim \text{Bern}(\pi_k)$.  
   c) Define $\alpha_i = s_i \circ z_i$ by an element-wise product.
   d) Sample noisy patch $R_i x \sim N(D\alpha_i, \gamma_i^{-1} I_P)$.

5) Construct the image $x$ as the average of all $R_i x$ that overlap on a given pixel.

non-convex problem. Using this definition of $h_i(x)$ in (1), a local optimal solution can be found by an alternating minimization procedure [32]: First solve the least squares solution for $x$ using the current values of $\alpha_i$ and $D$, and then update $\alpha_i$ and $D$, or only $\alpha_i$ if $D$ is learned off-line.

B. Dictionary Learning With Beta Process Factor Analysis

Typical dictionary learning approaches require a predefined dictionary size and, for each patch, the setting of either a sparsity level $T$, or an error threshold $\epsilon$ to determine how many dictionary elements are used. In both cases, if the settings do not agree with ground truth, the performance can significantly degrade. Instead, we consider a Bayesian nonparametric method called beta process factor analysis (BPFA) [23], which has been shown to successfully infer both of these values, as well as have competitive performance with algorithms in several application areas [23]–[26], and see [33]–[36] for related algorithms. The beta process is driven by an underlying Poisson process, and so it’s properties as a Bayesian nonparametric prior are well understood [27]. Originally used for survival analysis in the statistics literature, its use for latent factor modeling has been significantly increasing within the machine learning community [23]–[26], [28], [33]–[36].

1) Generative Model: We give the original hierarchical prior structure of the BPFA model in Algorithm 1, extending this to complex-valued dictionaries in Section III-A. With this approach, the model constructs a dictionary matrix $D \in \mathbb{C}^{P \times K}$ (with $\mathbb{C}$ denoting complex variables) of i.i.d. random variables, and assigns probability $\pi_k$ to vector $d_k$. The parameters for these probabilities are set such that most of the $\pi_k$ are expected to be small, with a few large. In Algorithm 1 we use an approximation to the beta process.\footnote{For a finite $c > 0$ and $\gamma > 0$, the random measure $H = \sum_{k=1}^{K} \pi_k \delta_{d_k}$ converges weakly to a beta process as $K \to \infty$ [24], [27].} Under this parameterization, each patch $R_i x$ extracted from the image $x$ is modeled as a sparse weighted combination of the dictionary elements, as determined by the element-wise product of $z_{ik} \in \{0, 1\}^K$ with the Gaussian vector $s_i$. What makes the model nonparametric is that for many values of $k$, the values of $z_{ik}$ will equal zero for all $i$ since $\pi_k$ will be very small; the model learns the number of these unused dictionary elements and their index values from the data. Therefore, the value of $K$ should be set to a large number that is more than the expected size of the dictionary. It can be shown that, under the assumptions of this prior, in the limit $K \to \infty$, the number of dictionary elements used by a patch is Poisson($\gamma$) distributed and the total number of dictionary elements used by the data grows like $c \gamma N$, where $N$ is the number of patches [28]. The parameters of the model include $c, \gamma, e_0, f_0, \theta_0, h_0$ and $K$; we discuss setting these values in Section IV.

2) Relationship to K-SVD: Another widely used dictionary learning method is K-SVD [20]. Though they are models for the same problem, BPFA and K-SVD have some significant differences that we briefly discuss. K-SVD learns the sparsity pattern of the coding vector $\alpha_i$ using the OMP algorithm [21] for each $i$. Holding the sparsity pattern fixed, it then updates each dictionary element and dimension of $\alpha$ jointly by a rank one approximation to the residual. Unlike BPFA, it learns as many dictionary elements as are given to it, so $K$ should be set wisely. BPFA on the other hand automatically prunes unneeded elements, and updates the sparsity pattern by using the posterior distribution of a Bernoulli process, which is significantly different from OMP. It updates the weights and the dictionary from their Gaussian posteriors as well. Because of this probabilistic structure, we derive a sampling algorithm for these variables that takes advantage of marginalization, and naturally learns the auxiliary variables $\gamma_i$ and $\gamma_s$.

3) Example Denoising Problem: As we will see, the relationship of dictionary learning to CS-MRI is essentially as a denoising step. To this end, we briefly illustrate BPFA on a denoising problem. Denoising of an image using dictionary learning proceeds by first learning the dictionary representation of each patch, $R_i x \approx D\alpha_i$. The denoised reconstruction of $x$ using BPFA is then $x_{\text{bpfa}} = \frac{1}{P} \sum_i R_i^T D\alpha_i$.

We show an example using 6 $\times$ 6 patches extracted from the noisy 512 $\times$ 512 image shown in Fig. 1(a). In Fig. 1(b) we show the resulting denoised image. For this problem we truncated the dictionary size to $K = 108$ and set all other model parameters to one. In Figs. 1(c) and 1(d) we show some statistics from dictionary learning. For example, Fig. 1(c) shows the values of $\pi_k$ sorted, where we see that fewer than 100 elements are used by the data, many of which are very sparsely used. Fig. 1(d) shows the empirical distribution of the number of elements used per patch. We see the ability of the model to adapt the sparsity to the complexity of the patch. In Table I we show PSNR results for three noise variance levels. For K-SVD, we consider the case when the error parameter matches the ground truth, and when it mismatches it by a magnitude of five. As expected, when K-SVD does not have an appropriate parameter setting the performance suffers. BPFA on the other hand adaptively infers this value, which helps improve the denoising.

III. CS-MRI With BPFA AND TV Penalty

We next present our approach for reconstructing single-channel MR images from highly undersampled $k$-space data.
In reference to the discussion in Section II, we consider a distribution on the number of dictionary elements used per patch. (c) Shows the final probabilities of the dictionary elements and (d) shows a distribution on the number of dictionary elements used per patch.

In this optimization. Since we sample a new value of $\gamma_c$ with each iteration of the algorithm discussed shortly, this trade-off is adaptively changing.

The total variation penalty $TV(x)$ we use is the isotropic TV model. Let $\psi_i$ be the $2 \times N$ difference operator for pixel $i$. Each row of $\psi_i$ contains a 1 centered on pixel $i$, and the first row also has a $-1$ on the pixel directly above pixel $i$, while the second has a $-1$ corresponding to the pixel to the right, and zeros elsewhere. Let $\Psi = [\psi_1, \ldots, \psi_N]^T$ be the resulting $2N \times N$ difference matrix for the entire image. The TV coefficients are $\beta = \Psi x \in \mathbb{C}^{2N}$, and the isotropic TV penalty is $TV(x) = \sum_i \|\psi_i x\|_2^2 = \sum_i |\beta_{2i-1}^2| + |\beta_{2i}^2|$, where $i$ ranges over the pixels in the MR image. For optimization we use the alternating direction method of multipliers (ADMM) [30, 31]. ADMM works by performing dual ascent on the augmented Lagrangian objective function introduced for the total variation coefficients. For completeness, we give a brief review of ADMM in the appendix.

### A. Algorithm

We present an algorithm for finding a local optimal solution to the non-convex objective function given in (5). We can write this objective as

$$L(x, \phi, \psi, \beta) = \sum_i \lambda_i \|\psi_i x\|_2^2 + \sum_i \gamma_i \|R_i x - Da_i\|_2^2 + \sum_i f(\varphi_i)$$

We seek to minimize this function with respect to $x$ and the dictionary learning variables $\varphi_i = \{D, s_i, z_i, \gamma_c, \gamma_s, \pi\}$.

Our first step is to put the objective into a more suitable form. We begin by defining the TV coefficients for the $i$th pixel as $\beta_i = [\beta_{2i-1}, \beta_{2i}]^T = \psi_i x$. We introduce the vector of Lagrange multipliers $\eta_i$, and then split $\beta_i$ from $\psi_i x$ by relaxing the equality via an augmented Lagrangian. This results in the objective function

$$L(x, \beta, \pi, \psi, \beta) = \sum_{i=1}^N \lambda_i \|\beta_i\|_2 + \eta_i f(\psi_i x - \beta_i) + \frac{\rho}{2} \|\psi_i x - \beta_i\|_2^2$$

$$+ \sum_{i=1}^N \gamma_i \|R_i x - Da_i\|_2^2 + f(\varphi_i)$$

$$+ \frac{\delta}{2} \|F_{ux} - y\|_2^2$$

From the ADMM theory [32], this objective will have (local) optimal values $\beta_i^* \pi^*$ and $x^*$. The ADMM theory can be split into three separate sub-problems: one for TV, one for BPFA, and one for updating the reconstruction $x$. Following the discussion of ADMM in the appendix, we define $u_i = (1/\rho) \eta_i$ and complete the square in the first line of (7). We then cycle through the following three sub-problems,

1. **(P1)** $\beta_i^* = \arg\min_{\beta} \lambda_i \|\beta\|_2 + \frac{\rho}{2} \|\psi_i x - \beta + u_i\|_2^2$,  
   $u_i = u_i + \psi_i x - \beta_i^*$,  
   $i = 1, \ldots, N$.

   For a fixed $D, s_i, z_i$ and $x$ the solution is also globally optimal.
Outline of Algorithm

Input: $y$ - Undersampled $k$-space data
Output: $x$ - Reconstructed MR image

Initialize: $x = F_H^H y$ and each $u_i = 0$. Sample $D$ from prior.

Step 1. P1: Optimize each $\beta_i$ via shrinkage.
Step 2. Update Lagrange multiplier vectors $u_i$.
Step 3. P2: Gibbs sample BPFA variables once.
Step 4. P3: Solve for $x$ using Fourier domain.

If not converged then return to Step 1.

(P2) $\varphi' = \arg \min \sum \frac{\gamma}{2} \| R_i x - D \alpha_i \|^2_2 + f(\varphi_i)$,

(P3) $x' = \arg \min \sum \frac{\rho}{2} \| \psi_i x - \beta_i' + u_i' \|^2_2 + \frac{\lambda}{2} \| F w - y \|^2_2$

Solutions for sub-problems P1 and P3 are globally optimal (conditioned on the most recent values of all other parameters). We cannot solve P2 analytically since the optimal values for the set of all BPFA variables do not have a closed form solution. Our approach for P2 is to use stochastic optimization by Gibbs sampling each variable of BPFA conditioned on current values of all other variables. We next present the updates for each sub-problem. We give an outline in Algorithm 2.

1) Algorithm for P1 (Total Variation): We can solve for $\beta_i$ exactly for each pixel $i = 1, \ldots, N$ by using a generalized shrinkage operation [31],

$$\beta_i' = \max \left\{ \frac{\psi_i x + u_i}{\| \psi_i x + u_i \|^2_2}, 0 \right\}.$$ (8)

We recall that $\beta_i'$ corresponds to the 2D TV coefficients for pixel $i$, with differences in one direction vertically and horizontally. We then update the corresponding Lagrange multiplier, $u_i = u_i + \psi_i x - \beta_i'$.

2) Algorithm for P2 (BPFA): We update the parameters of BPFA using Gibbs sampling. We are therefore stochastically optimizing (7), but only for this sub-problem. With reference to Algorithm 1, the P2 sub-problem entails sampling new values for the complex dictionary $D$, the binary vectors $z_i$ and real-valued weights $s_i$ (with which we construct $a_i = s_i \circ z_i$ through the element-wise product), the precisions $\gamma_e$ and $\gamma_s$, and the probabilities $\pi_{1:k}$, with $\pi_a$ giving the probability that $z_{ik} = 1$. In principle, there is no limit to the number of samples that can be made, with the final sample giving the updates used in the other sub-problems. We found that a single sample is sufficient in practice and leads to a faster algorithm. We describe the sampling procedure below.

a) Sample dictionary $D$: We define the $P \times N$ matrix $X = [R_1 x, \ldots, R_N x]$, which is a complex matrix of all vectorized patches extracted from the image $x$. We also define the $K \times N$ matrix $\alpha = [\alpha_1, \ldots, \alpha_N]$ containing the dictionary weight coefficients for the corresponding columns in $X$ such that $D \alpha$ is an approximation of $X$ to which we add noise from a circularly-symmetric complex normal distribution.

The update for the dictionary $D$ is

$$D = X \alpha^T (\alpha \alpha^T + (P/\gamma_e) I_p)^{-1} + E,$$ (9)

where $E_p = \text{ind} \mathcal{C} \mathcal{N}(0, (\gamma_e \alpha \alpha^T + P I_p)^{-1})$, $\alpha = 1, \ldots, P$.

When $z_{ik} = 1$, the mean of $s_{ik}$ is the regularized least squares solution and the variance will be small if $\gamma_e$ is large. When $z_{ik} = 0$, $s_{ik}$ can be sampled from the prior, but does not factor in the model in this case.

b) Sample sparse coding $a_i$: Sampling $a_i$ entails sampling $s_{ik}$ and $z_{ik}$ for each $k$. We sample these values using block sampling. We recall that to block sample two variables from their joint conditional posterior distribution, $(s, z) \sim \rho(s, z|\gamma_e)$, one can first sample $z$ from the marginal distribution, $z \sim \rho(z|\gamma_e)$, and then sample $s|z \sim \rho(s|z, \gamma_e)$ from the conditional distribution. The other sampling direction is possible as well, but for our problem sampling $z \rightarrow s|z$ is more efficient for finding a mode of the objective function.

We define $r_{1:k}$ to be the residual error in approximating the $i$th patch with the current values from BPFA minus the $k$th dictionary element, $r_{1:k} = R_i x - \sum_j d_j s_{ij} z_{ij}$. We then sample $z_{ik}$ from its conditional posterior Bernoulli distribution $z_{ik} \sim \pi_{1:k} 0 + q_{ik} 1$, where following a simplification,

$$p_{ik} \propto \pi_k \left( 1 + (\gamma_e/\gamma_s) d_k^H d_k \right)^{-\frac{1}{2}}$$ (10)

$$q_{ik} \propto 1 - \pi_k.$$ (11)

The symbol $H$ denotes the conjugate transpose. The probabilities can be obtained by dividing both of these terms by their sum. We observe that the probability that $z_{ik} = 1$ takes into account how well dictionary element $d_k$ correlates with the residual $r_{1:k}$. After sampling $z_{ik}$ we sample the corresponding weight $s_{ik}$ from its conditional posterior Gaussian distribution,

$$s_{ik} | z_{ik} \sim \mathcal{N} \left( \frac{z_{ik}}{\gamma_s/\gamma_e + d_k^H d_k}, \frac{1}{\gamma_s + \gamma_e z_{ik} d_k^H d_k} \right).$$ (12)

When $z_{ik} = 1$, the mean of $s_{ik}$ is the regularized least squares solution and the variance will be small if $\gamma_e$ is large. When $z_{ik} = 0$, $s_{ik}$ can be sampled from the prior, but does not factor in the model in this case.

c) Sample $\gamma_e$ and $\gamma_s$: We next sample from the conditional gamma posterior distribution of the noise precision and weight precision,

$$\gamma_e \sim \text{Gam} \left( g_0 + \frac{1}{2} P N, h_0 + \frac{1}{2} \sum_i \| R_i x - D \alpha_i \|^2_2 \right),$$ (13)

$$\gamma_s \sim \text{Gam} \left( e_0 + \frac{1}{2} \sum_i |s_{ik}|^2, f_0 + \frac{1}{2} \sum_i |z_{ik}|^2 s_{ik}^2 \right).$$ (14)

The expected value of each variable is the first term of the distribution divided by the second, which is close to the inverse of the average empirical error for $\gamma_e$. 

4) Sample $\pi_k$: Sample each $\pi_k$ from its conditional beta posterior distribution,

$$\pi_k \sim Beta\left(a_0 + \sum_{i=1}^{N} z_{ik}, b_0 + \sum_{i=1}^{N} (1 - z_{ik})\right). \quad (15)$$

The parameters to the beta distribution include counts of how many times dictionary element $d_i$ was used by a patch.

3) Algorithm for P3 (MRI Reconstruction): The final subproblem is to reconstruct the image $x$. Our approach takes advantage of the Fourier domain similar to other methods, see [14], [30]. The corresponding objective function is

$$\begin{align*}
\min_{x} & \quad \sum_{i=1}^{N} \frac{\beta}{\Psi_1} \|\psi_i x - \beta_i + u_i\|_2 + \frac{\gamma}{\Psi_1} \|R \times x - D \alpha_i\|_2^2 \\
& \quad + \frac{\lambda}{2} \|\mathcal{F} u x - y\|_2^2.
\end{align*}$$

Since this is a least squares problem, $x$ has a closed form solution that satisfies

$$\begin{align*}
x' &= \arg \min_{x} \sum_{i=1}^{N} \frac{\beta}{\Psi_1} \|\psi_i x - \beta_i + u_i\|_2 + \frac{\gamma}{\Psi_1} \|R \times x - D \alpha_i\|_2^2 \\
& \quad + \frac{\lambda}{2} \|\mathcal{F} u x - y\|_2^2.
\end{align*}$$

Since the left matrix is diagonal we can perform element-wise updating of the Fourier coefficients $\theta$,

$$\begin{align*}
\theta_i &= \frac{\rho \mathcal{F}_i \Psi^T (\beta - u) + \gamma \mathcal{P} \mathcal{F}_i \lambda_{BPFA} + \lambda \mathcal{F}_i \mathcal{F}_u^H y}{\rho \Lambda_i + \gamma \mathcal{P} + \lambda \mathcal{F}_i \mathcal{F}_u^H 1}. \quad (19)
\end{align*}$$

We observe that the rightmost term in the numerator and denominator equals zero if $i$ is not a measured $k$-space location. We invert $\theta$ via the inverse Fourier transform $\mathcal{F}^H$ to obtain the reconstructed MR image $x'$.

B. Discussion on $\lambda$

In noise-free compressed sensing, the fidelity term $\lambda$ can tend to infinity giving an equality constraint for the measured $k$-space values [1]. However, when $y$ is noisy the setting of $\lambda$ is critical for most CS-MRI algorithms since this parameter controls the level of denoising in the reconstructed image. We note that a feature of dictionary learning CS-MRI approaches is that $\lambda$ can still be set to a very large value, and so parameter selection isn’t necessary here. This is because a denoised version of the image is obtained through dictionary learning (2BPFA in this paper) and can be taken as the denoised reconstruction. In Equation (19), we observe that by setting $\lambda$ to a large value, we are effectively fixing the measured $k$-space values and using the $k$-space projection of BPFA and TV to fill in the missing values. The reconstruction $x$ will be noisy, but have artifacts due to sub-sampling removed. The output image $x_{BPFA}$ is a denoised version of $x$ using BPFA in essentially the same manner as in Section II-B3. Therefore, the quality of our algorithm depends largely on the quality of BPFA as an image denoising algorithm [25]. We show examples of this using synthetic and clinical data in Sections IV-B and IV-E.

IV. EXPERIMENTAL RESULTS

We evaluate the proposed algorithm on real-valued and complex-valued MRI, and on a synthetic phantom. We consider three sampling masks: 2D random sampling, Cartesian sampling with random phase encodes (1D random), and pseudo radial sampling.4 We show an example of each mask in Fig. 2. We consider a variety of sampling rates for each mask. As a performance measure we use PSNR, and also consider SSIM [37]. We compare with three other algorithms: Sparse MRI [3], which as discussed above is a

4We used codes referenced in [31], [8], and [10] to generate these masks.
5http://www.eecs.berkeley.edu/~mlustig/Software.html
combination of wavelets and total variation, DLMRI [14],\(^6\) which is a dictionary learning method based on K-SVD, and PBDW [15],\(^7\) which is patch-based method that uses directional wavelets and therefore places greater restrictions on the dictionary. We use the publicly available code for these algorithms indicated above and used the built-in parameter settings, or those indicated in the relevant papers. We also compare with the BPFA algorithm without using total variation by setting \(\lambda_g = 0.\)

### A. Set-Up

For all images, we extract 6 × 6 patches where each pixel defines the upper left corner of a patch and wrap around the image at the boundaries; we investigate different patch sizes later to show that this is a reasonable size. We initialize \(x\) by zero-filling in \(k\)-space. We use a dictionary with \(K = 108\) initial dictionary elements, recalling that the final number of dictionary elements will be smaller due to the sparse BPFA prior. If 108 is found to be too small, \(K\) can be increased with the result being a slower inference algorithm.\(^8\) We ran 1000 iterations and use the results of the last iteration.

For regularization parameters, we set the data fidelity term \(\lambda = 10^{100}\). We are therefore effectively requiring equality with the measured values of \(k\)-space and allowing BPFA to fill in the missing values, as well as give a denoised reconstruction, as discussed in Section III-B and highlighted below in Sections IV-B and IV-E. After trying several values, we also found \(\lambda_g = 10\) and \(\rho = 1000\) to give good results. We set the BPFA hyperparameters as \(c = \gamma = e_0 = f_0 = g_0 = h_0 = 1\). These settings result in a relatively non-informative prior given the amount of data we have. However, we note that our algorithm was robust to these values, since the data overwhelms these prior values when calculating posterior distributions.

### B. Experiments on a GE Phantom

We consider a noisy synthetic example to highlight the advantage of dictionary learning for CS-MRI. In Fig. 3 we show results on a 256 × 256 GE phantom with additive noise having standard deviation \(\sigma = 0.1\). In this experiment we use BPFA without TV to reconstruct the original image using 30% Cartesian sampling. We show the reconstruction using zero-filling in Fig. 3(a). Since \(\lambda = 10^{100}\), we see in Fig. 3(b) that BPFA essentially helps reconstruct the underlying noisy image for \(x\). However, using the denoising property of the BPFA model shown in Fig. 1, we obtain the denoised reconstruction of Fig. 3(c) by focusing on \(x_{BPFA}\) from Equation (16). This is in contrast with the best result we could obtain with TV in Fig. 3(d), which places the TV penalty on the reconstructed image. As discussed, for TV the setting of \(\lambda\) relative to \(\lambda_g\) is important. We set \(\lambda = 1\) and swept through \(\lambda_g \in (0, 0.15)\), showing the result with highest PSNR in Fig. 3(d). Similar to Fig. 1 we show statistics from the BPFA model in Figs. 3(e)-(g). We see that roughly 80 dictionary elements were used (the unused noisy elements in Fig. 3(e) are draws from the prior). We note that 2.28 elements were used on average by a patch given that at least one was used, which discounts the black regions.

### C. Experiments on Real-Valued (Synthetic) MRI

For our synthetic MRI experiments, we consider two publicly available real-valued 512 × 512 MRI\(^9\) of a shoulder and lumbar. We construct these problems by applying the relevant sampling mask to the projection of real-valued MRI

\(^6\)http://www.ifp.illinois.edu/~yoram/DLMRI-Lab/Documentation.html
\(^7\)http://www.quxiaobo.org/index_publications.html
\(^8\)As discussed in Section II-B, in theory \(K\) can be infinitely large.

\(^9\)www3.americanradiology.com/pls/web1/wwimggal.vmg/wwimggal.vmg
into k-space. Though using such real-valued MRI data may not reflect clinical reality, we include this idealized setting to provide a complete set of experiments similar to other papers [3], [14], [15]. We evaluate the performance of our algorithm using PSNR and compare with Sparse MRI [3], DLMRI [14] and PBDW [15]. Although the original data is real-valued, we learn complex dictionaries since the reconstructions are complex. We consider our algorithm with and without the total variation penalty, denoted BPFA+TV and BPFA, respectively.

We present the PSNR results for all sampling masks and rates in Tables II and III. From these values we see the competitive performance of the propose dictionary learning algorithm. We also see a slight improvement by the addition of the TV penalty. As expected, we observe that 2D random sampling produced the best results, followed by pseudo-radial and Cartesian sampling, which is due to their decreasing level of incoherence, with greater incoherence producing artifacts that are more noise-like [3]. Since BPFA is good at denoising images, the algorithm naturally performs well in this setting. In Figs. 4 and 5 we show the absolute value of the residuals of different algorithms using one experiment from each MRI. We see an improvement using the proposed method, which has more noise-like errors.

**D. Experiments on Complex-Valued MRI**

We also consider two clinically obtained complex-valued MRI: We use the T2-weighted brain MRI from [4], which is a 256 × 256 MRI of a healthy volunteer from a 3T Siemens Trio Tim MRI scanner using the T2-weighted turbo spin echo.
TABLE IV
PSNR/SSIM RESULTS FOR COMPLEX-VALUED BRAIN MRI AS A FUNCTION OF SAMPLING PERCENTAGE. SAMPLING MASKS INCLUDE CARTESIAN SAMPLING WITH RANDOM PHASE ENCODES, 2D RANDOM SAMPLING AND PSEUDO RADIAL SAMPLING

<table>
<thead>
<tr>
<th>Mask</th>
<th>Sample %</th>
<th>BPFA+TV</th>
<th>BPFA</th>
<th>DLMRI</th>
<th>Sparse MRI</th>
<th>PBDW</th>
<th>Zero-filling</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cartesian</td>
<td>25</td>
<td>35.62 / 0.951</td>
<td>34.86 / 0.948</td>
<td>29.90 / 0.812</td>
<td>25.29 / 0.696</td>
<td>34.69 / 0.935</td>
<td>24.13 / 0.591</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>38.64 / 0.968</td>
<td>37.70 / 0.965</td>
<td>31.54 / 0.849</td>
<td>26.16 / 0.745</td>
<td>37.24 / 0.957</td>
<td>24.55 / 0.614</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>39.36 / 0.972</td>
<td>38.87 / 0.971</td>
<td>32.35 / 0.865</td>
<td>27.35 / 0.795</td>
<td>37.90 / 0.963</td>
<td>24.94 / 0.616</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>41.09 / 0.977</td>
<td>40.45 / 0.976</td>
<td>33.60 / 0.876</td>
<td>29.82 / 0.845</td>
<td>39.23 / 0.969</td>
<td>26.28 / 0.667</td>
</tr>
<tr>
<td>Random</td>
<td>10</td>
<td>31.57 / 0.923</td>
<td>31.24 / 0.920</td>
<td>29.38 / 0.821</td>
<td>24.85 / 0.756</td>
<td>31.15 / 0.921</td>
<td>23.23 / 0.536</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>36.49 / 0.963</td>
<td>35.44 / 0.961</td>
<td>30.16 / 0.774</td>
<td>22.68 / 0.651</td>
<td>34.22 / 0.942</td>
<td>21.18 / 0.493</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>38.83 / 0.962</td>
<td>38.38 / 0.964</td>
<td>31.62 / 0.804</td>
<td>26.28 / 0.672</td>
<td>36.29 / 0.960</td>
<td>23.52 / 0.504</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>40.75 / 0.979</td>
<td>40.00 / 0.973</td>
<td>32.83 / 0.862</td>
<td>31.16 / 0.934</td>
<td>37.62 / 0.968</td>
<td>26.58 / 0.582</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>42.70 / 0.984</td>
<td>42.24 / 0.984</td>
<td>34.09 / 0.887</td>
<td>31.90 / 0.965</td>
<td>39.38 / 0.976</td>
<td>27.67 / 0.630</td>
</tr>
<tr>
<td>Radial</td>
<td>10</td>
<td>30.76 / 0.914</td>
<td>30.68 / 0.914</td>
<td>27.78 / 0.680</td>
<td>19.79 / 0.482</td>
<td>30.78 / 0.886</td>
<td>19.06 / 0.367</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>34.00 / 0.949</td>
<td>33.79 / 0.950</td>
<td>29.49 / 0.734</td>
<td>22.07 / 0.640</td>
<td>33.99 / 0.937</td>
<td>20.87 / 0.498</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>36.92 / 0.967</td>
<td>36.60 / 0.967</td>
<td>30.78 / 0.768</td>
<td>24.22 / 0.739</td>
<td>36.34 / 0.958</td>
<td>22.57 / 0.537</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>39.72 / 0.977</td>
<td>39.37 / 0.977</td>
<td>31.91 / 0.794</td>
<td>26.64 / 0.797</td>
<td>38.38 / 0.970</td>
<td>24.34 / 0.574</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>41.81 / 0.982</td>
<td>41.54 / 0.982</td>
<td>32.77 / 0.807</td>
<td>28.20 / 0.827</td>
<td>39.74 / 0.975</td>
<td>25.43 / 0.600</td>
</tr>
</tbody>
</table>

Fig. 6. Reconstruction results for 25% pseudo radial sampling of a complex-valued MRI of the brain. (a) Original. (b) BPFA+TV. (c) BPFA. (d) PBDW. (e) DLMRI. (f) PSNR vs iteration. (g) BPFA+TV error. (h) BPFA error. (i) PBDW error. (j) DLMRI error.

For the brain MRI experiment we use both PSNR and SSIM as performance measures. We show these values in Table IV for each algorithm, sampling mask and sampling rate. As with the synthetic MRI, we see that our algorithm performs competitively with the state-of-the-art. We also see the significant improvement of all algorithms over zero-filling. Example reconstructions are shown for each MRI dataset in Figs. 6 and 7. Also in Fig. 7 are PSNR values for the lemon MRI. We see from the absolute error residuals for these experiments that the BPFA algorithm learns a slightly finer detail structure compared with other algorithms, with the errors being more noise-like. We also show the PSNR of BPFA+TV and BPFA as a function of iteration. As is evident, the algorithm does not necessarily need all 1000 iterations, but performs competitively even in half that number.

E. Experiments in the Noisy Setting

The MRI we have considered thus far have been essentially noiseless. For some MRI machines this may be an unrealistic assumption. We continue our evaluation of noisy MRI begun with the toy GE phantom in Section IV-B by evaluating how our model performs on clinically obtained MRI with additive noise. We show BPFA results without TV to highlight the dictionary learning features, but note that results with TV provide a slight improvement in terms of PSNR and
Fig. 7. Reconstruction results for 35% 2D random sampling of a complex-valued MRI of a lemon. (a) Original. (b) BPFA+TV: PSNR = 39.64. (c) BPFA: PSNR = 38.21. (d) PBDW: PSNR = 37.89. (e) DLMRI: PSNR = 35.05. (f) PSNR vs iteration. (g) BPFA+TV error. (h) BPFA error. (i) PBDW error. (j) DLMRI error.

Fig. 8. PSNR vs $\lambda$ in the noisy setting ($\sigma = 0.03$) for the complex-value brain MRI with 30% 2D random sampling. SSIM. We again consider the brain MRI and use additive complex white Gaussian noise having standard deviation $\sigma = 0.01, 0.02, 0.03$. For all experiments we use the original noise-free MRI as the ground truth.

As discussed in Section III-B and illustrated in Section IV-B, dictionary learning allows us to consider two possible reconstructions: the actual reconstruction $x$, and the denoised BPFA reconstruction $x_{BPFA} = \frac{1}{p} \sum_i R_i^T D_{ai}$. As detailed in these sections, as $\lambda$ becomes larger the reconstruction will be noisier, but with the artifacts from sub-sampling removed. However, for all values of $\lambda$, $x_{BPFA}$ produces a denoised version that essentially doesn’t change. We see this clearly in Fig. 8, where we show the PSNR of each reconstruction as a function of $\lambda$. When $\lambda$ is small, the performance degrades for both algorithms since too much smoothing is done by dictionary learning on $x$. As $\lambda$ increases, both improve, but eventually the reconstruction of $x$ degrades again because near equality to the noisy $y$ is being more strictly enforced. The denoised reconstruction however levels off and does not degrade. We show PSNR values in Table V as a function of noise level.

Table V

<table>
<thead>
<tr>
<th>PSNR for 35% Cartesian Sampling of Complex-Valued Brain MRI for Various Noise Standard Deviations ($\lambda = 10^{100}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconstruction method</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>BPFA–reconstruction</td>
</tr>
<tr>
<td>BPFA–denoising</td>
</tr>
<tr>
<td>DLMRI</td>
</tr>
</tbody>
</table>

Example reconstructions that parallel those given in Fig. 3 are also shown in Fig. 9. These results highlight the robustness of our approach to $\lambda$ in the noisy setting, and we note that we...
encountered no stability issues using extremely large values of $\lambda$.

\section*{F. Dictionary Learning and Further Discussion}

We investigate the model learned by BPFA. In Fig. 10 we show dictionary learning results learned by BPFA+TV for radial sampling of the complex Brain MRI. In the top portion, we show the dictionaries learned for 10\%, 20\% and 30\% sampling. We see that they are similar in their shape, but the number of elements increases as the sampling percentage increases since more complex information about the image is contained in the $k$-space measurements. We again note that unused elements are represented by draws from the prior. In Fig. 10(d) we show the cumulative sum of the ordered $\pi_k$ from BPFA. We can read off the average number of elements used per patch by looking at the right-most value. We see that more elements are used per patch as the fraction of observed $k$-space increases. We also see that for 10\%, 20\% and 30\% sampling, roughly 70, 80 and 95, respectively, of the 108 total dictionary elements were significantly used, as indicated by the leveling off of these functions. This highlights the adaptive property of the nonparametric beta process prior.

In Fig. 10(e) we show the empirical distribution on the number of dictionary elements used per patch for each sampling rate.

We see that there are two modes, one for the empty background and one for the foreground, and the second mode tends to increase as the sampling rate increases. The adaptability of this value to each patch is another characteristic of the beta process model.

We also performed an experiment with varying patch sizes and show our results in Table VI. We see that the results are not very sensitive to this setting and that comparisons using $6 \times 6$ patches are meaningful. We also compare the runtime for different algorithms in Table VII, showing both the total runtime of each algorithm and the per-iteration times using an Intel Xeon CPU E5-1620 at 3.60GHz, 16.0G ram. However, we note that we arguably ran more iterations than necessary for these algorithms; the BPFA algorithms generally produced high quality results in half the number of iterations, as did DLMRI (the authors of [14] recommend 20 iterations), while Sparse MRI uses 5 iterations as default and the performance didn’t improve beyond 10 iterations. We note that the speed-up over DLMRI arises from the lack of the OMP algorithm, which in Matlab is much slower than our sparse coding update. We note that inference for the BPFA model is easily parallelizable—as are the other dictionary learning algorithms—which can speed up processing time.

The proposed method has several advantages, which we believe leads to the improvement in performance. A significant advantage is the adaptive learning of the dictionary size and per-patch sparsity level using a nonparametric stochastic process that is naturally suited for this problem. Several other dictionary learning parameters such as the noise variance and the variances of the score weights are adjusted as well through a natural MCMC sampling approach. These benefits have been investigated in other applications of this model [25], and naturally translate here since CS-MRI with BPFA is closely related to image denoising as we have shown.

Another advantage of our model is the Markov Chain Monte Carlo inference algorithm itself. In highly non-convex Bayesian models (or similar models with a Bayesian interpretation), it is often observed by the statistics community that

\begin{table}[h]
\centering
\caption{PSNR as a Function of Patch Size for a Real-Valued and Complex-Valued Brain MRI with Cartesian Sampling}
\begin{tabular}{|c|c|c|c|c|}
\hline
Patch Size & 4x4 & 5x5 & 6x6 & 7x7 & 8x8 \\
\hline
\hline
Synthetic brain 25\% & 37.86 & 38.29 & 38.33 & 38.26 & 38.24 \\
Complex brain 40\% & 40.53 & 40.84 & 41.09 & 41.11 & 41.15 \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Total Runtime in Minutes (Seconds/Iteration). We Ran 1000 Iterations of BPFA, 100 of DLMRI and 10 of Sparse MRI}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Sampling % & BPFA+TV & BPFA & DLMRI & Sparse MRI \\
\hline
\hline
10\% & 52.4 (3.15) & 50.5 (3.03) & 27.6 (16.5) & 1.63 (9.78) \\
20\% & 51.3 (3.08) & 49.5 (2.97) & 38.3 (23.0) & 1.59 (9.54) \\
30\% & 51.2 (3.07) & 48.3 (2.90) & 45.7 (27.4) & 1.60 (9.60) \\
\hline
\end{tabular}
\end{table}

BPFA is significantly faster than K-SVD in Matlab because it requires fewer loops. This difference may not be as large with other coding languages.
MCMC sampling can outperform deterministic methods, and rarely performs worse [38]. Given that BPFA is a Bayesian model, such sampling techniques are readily derived, as we showed in Section III-A.

V. CONCLUSION

We have presented an algorithm for CS-MRI reconstruction that uses Bayesian nonparametric dictionary learning. Our Bayesian approach uses a model called beta process factor analysis (BPFA) for in situ dictionary learning. Through this hierarchical generative structure, we can learn the dictionary size, sparsity pattern and additional regularization parameters. We also considered a total variation penalty term for additional constraints on image smoothness. We presented an optimization algorithm using the alternating direction method of multipliers (ADMM) and MCMC Gibbs sampling for all BPFA variables. Experimental results on real and complex-valued MRI showed that our proposed regularization framework compares favorably with other algorithms for various sampling trajectories and rates. We also showed the natural ability of dictionary learning to handle noisy MRI without dependence on the measurement fidelity parameter \( \lambda \). To this end, we showed that the model can enforce a near equality constraint to the noisy measurements and use the dictionary learning result as a denoised output of the noisy MRI.

APPENDIX

We give a brief review of the ADMM algorithm [32]. We start with the convex optimization problem

\[
\min_x \|Ax - b\|_2^2 + h(x),
\]  

(20)

where \( h \) is a non-smooth convex function, such as an \( \ell_1 \) penalty. ADMM decouples the smooth squared error term from this penalty by introducing a second vector \( v \) such that

\[
\min_x \|Ax - b\|_2^2 + h(v) \quad \text{subject to} \quad v = x.
\]

(21)

This is followed by a relaxation of the equality \( v = x \) via an augmented Lagrangian term

\[
L(x, v, \eta) = \|Ax - b\|_2^2 + h(v) + \eta^T (x - v) + \frac{\rho}{2} \|x - v\|_2^2.
\]

(22)

A minimax saddle point is found with the minimization taking place over both \( x \) and \( v \) and dual ascent for \( \eta \).

Another way to write the objective in (22) is to define \( u = (1/\rho)\eta \) and combine the last two terms. The result is an objective that can be optimized by cycling through the following updates for \( x, v \), and \( u \),

\[
x' = \arg \min_x \|Ax - b\|_2^2 + \frac{\rho}{2} \|x - v + u\|_2^2,
\]

(23)

\[
v' = \arg \min_v h(v) + \frac{\rho}{2} \|x' - v + u\|_2^2,
\]

(24)

\[
u' = u + x' - v'.
\]

(25)

This algorithm simplifies the optimization since the objective for \( x \) is quadratic and thus has a simple analytic solution, while the update for \( v \) is a proximity operator of \( h \) with penalty \( \rho \), the difference being that \( v \) is not pre-multiplied by a matrix as \( x \) is in (20). Such objective functions tend to be easier to optimize. For example when \( h \) is the TV penalty the solution for \( v \) is analytical.

ACKNOWLEDGMENT

Y. Huang and J. Paisley contributed equally to this work.

REFERENCES


