Iterative Reweighted ℓ_1 and ℓ_2 Methods for Finding Sparse Solutions

David Wipf and Srikantan Nagarajan

***Abstract*—A variety of practical methods have recently been introduced for finding maximally sparse representations from overcomplete dictionaries, a central computational task in compressive sensing applications as well as numerous others. Many of the underlying algorithms rely on iterative reweighting schemes that produce more focal estimates as optimization progresses. Two such variants are iterative reweighted ℓ_1 and ℓ_2 minimization; however, some properties related to convergence and sparse estimation, as well as possible generalizations, are still not clearly understood or fully exploited. In this paper, we make the distinction between *separable* and *non-separable* iterative reweighting algorithms. The vast majority of existing methods are separable, meaning the weighting of a given coefficient at each iteration is only a function of that individual coefficient from the previous iteration (as opposed to dependency on all coefficients). We examine two such separable reweighting schemes: an ℓ_2 method from Chartrand and Yin (2008) and an ℓ_1 approach from Candès *et al.* (2008), elaborating on convergence results and explicit connections between them. We then explore an interesting non-separable alternative that can be implemented via either ℓ_2 or ℓ_1 reweighting and maintains several desirable properties relevant to sparse recovery despite a highly non-convex underlying cost function. For example, in the context of canonical sparse estimation problems, we prove uniform superiority of this method over the minimum ℓ_1 solution in that, 1) it can never do worse when implemented with reweighted ℓ_1, and 2) for any dictionary and sparsity profile, there will always exist cases where it does better. These results challenge the prevailing reliance on strictly convex (and separable) penalty functions for finding sparse solutions. We then derive a new non-separable variant with similar properties that exhibits further performance improvements in empirical tests. Finally, we address natural extensions to group sparsity problems and non-negative sparse coding.**

***Index Terms*—Compressive sensing (CS), iterative reweighting algorithms, sparse representations, underdetermined inverse problems.**

I. Introduction

With the advent of compressive sensing and other related applications, there has been growing interest in finding sparse signal representations from redundant dictionaries [7], [8], [12], [13], [24]. The canonical form of this problem is given by

$$\min_{\boldsymbol{x}} ||\boldsymbol{x}||_0, \quad \text{s.t. } \boldsymbol{y} = \Phi \boldsymbol{x} \tag{1}$$

where $\Phi \in \mathbb{R}^{n \times m}$ is a matrix whose columns $\boldsymbol{\phi}_i$ represent an overcomplete or redundant basis (i.e., $\text{rank}(\Phi) = n$ and $m > n$), $\boldsymbol{x} \in \mathbb{R}^m$ is a vector of unknown coefficients to be learned, and $\boldsymbol{y}$ is the signal vector. The cost function being minimized represents the ℓ_0 norm of $\boldsymbol{x}$, which is a count of the nonzero elements in $\boldsymbol{x}$.[1] If measurement noise or modeling errors are present, we instead solve the alternative problem

$$\min_{\boldsymbol{x}} ||\boldsymbol{y} - \Phi \boldsymbol{x}||_2^2 + \lambda ||\boldsymbol{x}||_0, \quad \lambda > 0 \tag{2}$$

noting that in the limit as $\lambda \to 0$, the two problems are equivalent (the limit must be taken outside of the minimization).

Unfortunately, an exhaustive search for the optimal representation requires the solution of up to $\binom{m}{n}$ linear systems of size $n \times n$, a prohibitively expensive procedure for even modest values of m and n. Consequently, in practical situations there is a need for approximate methods that efficiently solve (1) or (2) in most cases. Many recent sparse approximation algorithms rely on iterative reweighting schemes that produce more focal estimates as optimization progresses [3]–[5], [11], [17], [18]. Two such variants are iterative reweighted ℓ_2 and ℓ_1 minimization. For the former, at the $(k+1)$th iteration we must compute

$$\begin{aligned} \boldsymbol{x}^{(k+1)} &\to \arg\min_{\boldsymbol{x}} ||\boldsymbol{y} - \Phi \boldsymbol{x}||_2^2 + \lambda \sum_i w_i^{(k)} x_i^2 \\ &= \tilde{W}^{(k)} \Phi^T \left(\lambda I + \Phi \tilde{W}^{(k)} \Phi^T \right)^{-1} \boldsymbol{y} \end{aligned} \tag{3}$$

where $\tilde{W}^{(k)}$ is a diagonal weighting matrix from the kth iteration with ith diagonal element $1/w_i^{(k)}$ that is potentially a function of all $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(k)}$. Similarly, the ℓ_1 reweighting variant solves

$$\boldsymbol{x}^{(k+1)} \to \arg\min_{\boldsymbol{x}} ||\boldsymbol{y} - \Phi \boldsymbol{x}||_2^2 + \lambda \sum_i w_i^{(k)} |x_i| \tag{4}$$

although no analytic solution exists in this case and so a numerical program (e.g., interior point method) must be adopted. While this is typically an expensive computation, the number of iterations of (4) required is generally much less than (3) and, unlike (3), even a single iteration produces a sparse solution. A second advantage of (4) is that it is often much easier to

Manuscript received February 20, 2009; revised October 31, 2009. Current version published March 17, 2010. This work was supported by the National Institute of Health (NIH) under Grants R01DC004855 and R01 DC006435 and the NIH/NCRR UCSF-CTSI under Grant UL1 RR024131. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Rick Chartrand.

The authors are with the Biomagnetic Imaging Laboratory, University of California, San Francisco, CA 94143 USA (e-mail: davidwipf@gmail.com; sri@radiology.ucsf.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2010.2042413

[1]Note that $||\boldsymbol{x}||_0$, because it does not satisfy the required axioms, is not technically a norm.

incorporate additional constraints, e.g., bounded activation or non-negativity of $\boldsymbol{x}$ [2]. This is because few iterations are needed and the per-iteration computational complexity need not change significantly with the inclusion of many useful constraints. In contrast, with (3), we lose the advantage of closed-form updates on $\boldsymbol{x}$ when such constraints are imposed. While we could still easily solve (3) numerically as we do with (4), the large number of iterations required renders this approach much less attractive. In general, it is possible to perform iterative reweighting with an arbitrary convex function of the coefficients $f(\boldsymbol{x})$. However, ℓ_1 and ℓ_2 norms are typically selected since the former is the sparsest convex penalty (which means it results in the tightest convex bound to potentially non-convex underlying penalty functions), while the latter gives closed form solutions unlike virtually any other useful function.

In both ℓ_2 and ℓ_1 reweighting schemes, different methods are distinguished by the choice of $\boldsymbol{w}^{(k)} \triangleq [w_1^{(k)}, \ldots, w_m^{(k)}]^T$, which ultimately determines the surrogate cost function for promoting sparsity that is being minimized (although not all choices lead to convergence or sparsity). In this paper, we will analyze several different weighting selections, examining convergence issues, analytic properties related to sparsity, and connections between various algorithms. In particular, we will discuss a central dichotomy between what we will refer to as *separable* and *non-separable* choices for $\boldsymbol{w}^{(k)}$. By far the more common, separable reweighting implies that each $w_i^{(k)}$ is only a function of $x_i^{(k)}$. This scenario is addressed in Section II where we analyze the ℓ_2 method from Chartrand and Yin [4] and an ℓ_1 approach from Candès *et al.* [3], elaborating on convergence results and explicit connections between them.

In contrast, the non-separable case means that each $w_i^{(k)}$ is a function of all $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(k)}$, i.e., it is potentially dependent on all coefficients from all past iterations. Section III explores an interesting non-separable alternative, based on a dual-form interpretation of sparse Bayesian learning (SBL) [23], [27], that can be implemented via either ℓ_2 or ℓ_1 reweighting, with the former leading to a direct connection with Chartrand and Yin's algorithm. Overall, it maintains several desirable properties relevant to sparse recovery despite a highly non-convex underlying cost function. For example, in the context of (1), we prove uniform superiority of this method over the minimum ℓ_1 solution, which represents the best convex approximation, in that 1) it can never do worse when implemented with reweighted ℓ_1, and 2) for any Φ and sparsity profile, there will always exist cases where it does better. To a large extent, this removes the stigma commonly associated with using non-convex sparsity penalties. Later in Section III-C, we derive a second promising non-separable variant by starting with a plausible ℓ_1 reweighting scheme and then working backwards to determine the form and properties of the underlying cost function.

Section IV then explores extensions of iterative reweighting algorithms to areas such as the simultaneous sparse approximation problem [19], [25], which is a special case of covariance component estimation [27]. In this scenario we are presented with multiple signals $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots$ that we assume were produced by coefficient vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ characterized by the same sparsity profile or support. All of the algorithms discussed herein can naturally be expanded to this domain by applying an appropriate penalty to the aligned elements of each coefficient vector. Finally, simulations involving all of these methods are contained in Section V, with brief concluding remarks in Section VI. Portions of this work were presented at the *Workshop on Signal Processing with Adaptive Sparse/Structured Representations* in Saint-Malo, France, 2009 and at the *Advances in Neural Information Processing Systems Conference*, Vancouver, BC, Canada, 2009.

II. Separable Reweighting Schemes

Separable reweighting methods have been applied to sparse recovery problems both in the context of the ℓ_2 norm [4], [10], [17], [18], and more recently the ℓ_1 norm [3], [11]. All of these methods (at least locally) attempt to solve

$$\min_{\boldsymbol{x}} \sum_i g(x_i), \quad \text{s.t. } \boldsymbol{y} = \Phi \boldsymbol{x} \tag{5}$$

or

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \Phi \boldsymbol{x}\|_2^2 + \lambda \sum_i g(x_i) \tag{6}$$

where $g(x_i)$ is a non-decreasing function of $|x_i|$. In general, if $g(x_i)$ is concave in x_i^2, it can be handled using reweighted ℓ_2 [17], while any $g(x_i)$ that is concave in $|x_i|$ can be used with reweighted ℓ_1 [9]. Many of these methods have been analyzed extensively in the past; consequently, we will briefly address outstanding issues pertaining to two new approaches with substantial promise.

A. ℓ_2 Reweighting Method of Chartrand and Yin [4]

In [4], the ℓ_2 reweighting

$$w_i^{(k+1)} \rightarrow \left[\left(x_i^{(k+1)} \right)^2 + \epsilon^{(k+1)} \right]^{-1} \tag{7}$$

is proposed (among others that are shown empirically to be less successful as discussed below), where $\epsilon^{(k+1)} \geq 0$ is a regularization factor that is reduced to zero as k becomes large. This procedure leads to state-of-the-art performance recovering sparse solutions in a series of empirical tests using a heuristic for updating $\epsilon^{(k+1)}$ and assuming $\lambda \rightarrow 0$ (noiseless case); however, no formal convergence proof is provided. Here, we sketch a proof that this method will converge, for arbitrary sequences $\epsilon^{(k+1)} \rightarrow 0$, to a local minimum of a close surrogate cost function to (1) (similar ideas apply to the more general case where λ is nonzero).

To begin, we note that there is a one-to-one correspondence between minima of (1) and minima of

$$\min_{\boldsymbol{x}} \sum_i \log |x_i|, \quad \text{s.t. } \boldsymbol{y} = \Phi \boldsymbol{x} \tag{8}$$

which follows since $\|\boldsymbol{x}\|_0 \equiv \lim_{p \rightarrow 0} \sum_i |x_i|^p$ and $\lim_{p \rightarrow 0} (1/p) \sum_i (|x_i|^p - 1) = \sum_i \log |x_i|$. This log-based penalty function is frequently used for finding sparse solutions in signal and image processing using iterative locally minimizing procedures (e.g., see [10] and [18]).[2]

[2]To be precise regarding well-defined local minima, we could add a small ϵ regularizer within the logarithm of (8) and then take the limit (outside of the minimization) as this ϵ goes to zero, but the analysis which follows (and the associated reweighted ℓ_2 updates for minimization) is all essentially the same.

The penalty function in (8) can be upper-bounded using

$$\begin{aligned}\log|x_i| &\leq \frac{1}{2}\log|x_i^2+\epsilon| \\ &\leq \frac{x_i^2+\epsilon}{2\gamma_i}+\frac{1}{2}\log\gamma_i-\frac{1}{2} \end{aligned} \tag{9}$$

where $\epsilon, \gamma_i \geq 0$ are arbitrary. The second inequality, which follows directly from the concavity of the log function with respect to x_i^2, becomes an equality iff $\gamma_i = x_i^2+\epsilon$. Now consider solving

$$\min_{\boldsymbol{x},\boldsymbol{\gamma},\epsilon} \sum_i \left(\frac{x_i^2+\epsilon}{\gamma_i}+\log\gamma_i\right), \quad \text{s.t.} \quad \boldsymbol{y}=\Phi\boldsymbol{x}, \quad \epsilon\geq 0, \quad \gamma_i \geq 0, \quad \forall i \tag{10}$$

where $\boldsymbol{\gamma} \triangleq [\gamma_1,\ldots,\gamma_m]^T$. Again, there is a one-to-one correspondence between local minima of the original problem (1) and local minima of (10). For fixed ϵ and $\boldsymbol{\gamma}$, the optimal $\boldsymbol{x}$ satisfies $\boldsymbol{x}=\Gamma^{1/2}(\Phi\Gamma^{1/2})^{\dagger}\boldsymbol{y}$, with $\Gamma \triangleq \mathrm{diag}[\boldsymbol{\gamma}]$ and $(\cdot)^{\dagger}$ denoting the Moore–Penrose pseudo-inverse. From above, the minimizing $\boldsymbol{\gamma}$ for fixed $\boldsymbol{x}$ and ϵ is $\gamma_i = x_i^2+\epsilon, \forall i$. Consequently, coordinate descent over $\boldsymbol{x}$, $\boldsymbol{\gamma}$, and ϵ is guaranteed to reduce or leave unchanged (10) (the exact strategy for reducing ϵ is not crucial).

It can be shown that these updates, which are equivalent to Chartrand and Yin's ℓ_2 reweighting algorithm with $w_i^{(k)}=\gamma_i^{-1}$, are guaranteed to converge monotonically to a local minimum (or saddle point) of (10) by satisfying the conditions of the *Global Convergence Theorem* (see for example [31]). At any such solution, $\epsilon=0$ and we will also be at a local minimum (or saddle point) of (8). In the unlikely event that a saddle point is reached (such solutions to (8) are very rare [20]), a small perturbation leads to a local minimum.

Of course obviously we could set $\epsilon \to 0$ in the very first iteration, which reproduces the FOCUSS algorithm and its attendant quadratic rate of convergence near a minimum [20]; however, compelling evidence from [4] suggests that slow reduction of ϵ is far more effective in avoiding suboptimal local minima troubles.

The weight update (7) is part of a wider class given by

$$w_i^{(k+1)} \to \left[\left(x_i^{(k+1)}\right)^2+\epsilon^{(k+1)}\right]^{\frac{p}{2}-1} \tag{11}$$

where $0 \leq p \leq 2$ is a user-defined parameter. With $p=0$, we recover (7) and also obtain the best empirical performance solving (1) according to experiments in [4]; other values for p lead to alternative implicit cost functions and convergence properties. Additionally, for a carefully chosen $\epsilon^{(k+1)}$ update, interesting and detailed convergence results are possible using (11), particularly for the special case where $p=1$, which produces a robust means of finding a minimum ℓ_1 norm solution using reweighted ℓ_2 [6]. However, the selection for $\epsilon^{(k+1)}$ used to obtain these results may be suboptimal in certain situations relative to other prescriptions for choosing p and ϵ (see Section V-A).[3] Regardless, the underlying analysis from [6] provides useful insights into reweighted ℓ_2 algorithms.

[3]Note that the convergence analysis we discuss above applies for any sequence of $\epsilon^{(k)} \to 0$ and can be extended to other values of $p \in [0,1)$.

B. ℓ_1 Reweighting Method of Candès et al. [3]

An interesting example of separable iterative ℓ_1 reweighting is presented in [3] where the selection

$$w_i^{(k+1)} \to \left[\left|x_i^{(k+1)}\right|+\epsilon\right]^{-1} \tag{12}$$

is suggested. Here, ϵ is generally chosen as a fixed, application-dependent constant. In the noiseless case, it is demonstrated based on [9] that this amounts to iteratively solving

$$\min_{\boldsymbol{x}} \sum_i \log(|x_i|+\epsilon), \quad \text{s.t. } \boldsymbol{y}=\Phi\boldsymbol{x} \tag{13}$$

and convergence to a local minimum or saddle point is guaranteed. The FOCUSS algorithm [20] can also be viewed as an iterative reweighted ℓ_2 method for locally solving (13) for the special case when $\epsilon=0$; however, Candès *et al.* point out that just a few iterations of their method is far more effective in finding sparse solutions than FOCUSS. This occurs because, with $\epsilon=0$, the cost function (13) has on the order of $\binom{m}{n}$ deep local minima and so convergence to a suboptimal one is highly likely. Here we present reweighted ℓ_2 updates for minimizing (13) for arbitrary ϵ, which allows direct comparisons between reweighted ℓ_1 and ℓ_2 on the same underlying cost function (which is not done in [3]). Using results from [17], it is relatively straightforward to show that

$$\begin{aligned}\log(|x_i|+\epsilon) \leq{}& \frac{x_i^2}{\gamma_i} \\ &+\log\left[\frac{\left(\epsilon^2+2\gamma_i\right)^{\frac{1}{2}}+\epsilon}{2}\right]-\frac{\left[\left(\epsilon^2+2\gamma_i\right)^{\frac{1}{2}}-\epsilon\right]^2}{4\gamma_i}\end{aligned} \tag{14}$$

for all $\epsilon, \gamma_i \geq 0$, with equality iff $\gamma_i = x_i^2+\epsilon|x_i|$. Note that this is a very different surrogate cost function than (9) and utilizes the concavity of $\log(|x_i|+\epsilon)$, as opposed to $\log(x_i^2+\epsilon)$, with respect to x_i^2. Using coordinate descent as before with $w_i^{(k)}=\gamma_i^{-1}$ leads to the reweighted ℓ_2 iteration

$$w_i^{(k+1)} \to \left[\left(x_i^{(k+1)}\right)^2+\epsilon\left|x_i^{(k+1)}\right|\right]^{-1} \tag{15}$$

which will reduce (or leave unchanged) (13) for arbitrary $\epsilon \geq 0$. In preliminary empirical tests, this method is superior to regular FOCUSS and could be used as an alternative to reweighted ℓ_1 if computational resources for computing ℓ_1 solutions are limited. Additionally, as stated above the most direct comparison between reweighted ℓ_1 and ℓ_2 in this context would involve empirical tests using (15) versus the method from Candès *et al.*, which we explore in Section V-C. It would also be worthwhile to compare (15) using a decreasing ϵ update with (7) since both are derived from different implicit bounds on $\log|x_i|$.

III. Non-Separable Reweighting Schemes

Non-separable selections for $\boldsymbol{w}^{(k)}$ allow us to minimize cost functions based on general, non-separable sparsity penalties, meaning penalties that cannot be expressed as a summation over functions of the individual coefficients as in (6). Such penalties potentially have a number of desirable properties [28]. In this

section, we analyze three non-separable reweighting strategies. The first two are based on a dual-space sparse Bayesian learning (SBL); the third is derived based on simple intuitions gained from working with non-separable algorithms.

A. ℓ_2 Reweighting Applied to a Dual-Space SBL Penalty

A particularly useful non-separable penalty emerges from a dual-space view [27] of SBL [23], which is based on the notion of automatic relevance determination (ARD) [16]. SBL assumes a Gaussian likelihood function $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}; \Phi\boldsymbol{x}, \lambda I)$, consistent with the data fit term from (2). The basic ARD prior incorporated by SBL is $p(\boldsymbol{x};\boldsymbol{\gamma}) = \mathcal{N}(\boldsymbol{x}; 0, \mathrm{diag}[\boldsymbol{\gamma}])$, where $\boldsymbol{\gamma} \in \mathbb{R}^m_+$ is a vector of m non-negative hyperparameters governing the prior variance of each unknown coefficient. These hyperparameters are estimated from the data by first marginalizing over the coefficients $\boldsymbol{x}$ and then performing what is commonly referred to as evidence maximization or type-II maximum likelihood [16], [23]. Mathematically, this is equivalent to minimizing

$$\begin{aligned}\mathcal{L}(\boldsymbol{\gamma}) &\triangleq -\log \int p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x};\boldsymbol{\gamma})d\boldsymbol{x} \\ &= -\log p(\boldsymbol{y};\boldsymbol{\gamma}) \\ &\equiv \log|\Sigma_y| + \boldsymbol{y}^T\Sigma_y^{-1}\boldsymbol{y} \end{aligned} \tag{16}$$

where $\Sigma_y \triangleq \lambda I + \Phi\Gamma\Phi^T$ and $\Gamma \triangleq \mathrm{diag}[\boldsymbol{\gamma}]$. Once some $\boldsymbol{\gamma}_* = \arg\min_{\boldsymbol{\gamma}} \mathcal{L}(\boldsymbol{\gamma})$ is computed, an estimate of the unknown coefficients can be obtained by setting $\boldsymbol{x}_{\mathrm{SBL}}$ to the posterior mean computed using $\boldsymbol{\gamma}_*$

$$\boldsymbol{x}_{\mathrm{SBL}} = \mathrm{E}[\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\gamma}_*] = \Gamma_*\Phi^T\Sigma_{y*}^{-1}\boldsymbol{y}. \tag{17}$$

Note that if any $\gamma_{*,i} = 0$, as often occurs during the learning process, then $x_{\mathrm{SBL},i} = 0$ and the corresponding dictionary column is effectively pruned from the model. The resulting $\boldsymbol{x}_{\mathrm{SBL}}$ is therefore sparse, with nonzero elements corresponding with the "relevant" basis vectors.

It is not immediately apparent how the SBL procedure, which requires optimizing a cost function in $\boldsymbol{\gamma}$-space and is based on a separable prior $p(\boldsymbol{x};\boldsymbol{\gamma})$, relates to solving/approximating (1) and/or (2) via a non-separable penalty in $\boldsymbol{x}$-space. However, it has been shown in [27] that $\boldsymbol{x}_{\mathrm{SBL}}$ satisfies

$$\boldsymbol{x}_{\mathrm{SBL}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \Phi\boldsymbol{x}\|_2^2 + \lambda g_{\mathrm{SBL}}(\boldsymbol{x}) \tag{18}$$

where

$$g_{\mathrm{SBL}}(\boldsymbol{x}) \triangleq \min_{\boldsymbol{\gamma}\geq 0} \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \log|\alpha I + \Phi\Gamma\Phi^T| \tag{19}$$

assuming $\alpha = \lambda$. While not discussed in [27], $g_{\mathrm{SBL}}(\boldsymbol{x})$ is a general penalty function that only need have $\alpha = \lambda$ to obtain equivalence with SBL; other selections may lead to better performance.

The analysis in [27] reveals that replacing $\|\boldsymbol{x}\|_0$ with $g_{\mathrm{SBL}}(\boldsymbol{x})$ and $\alpha \to 0$ leaves the globally minimizing solution to (1) unchanged but drastically reduces the number of local minima (more so than *any possible* separable penalty function). While details are beyond the scope of this paper, these ideas can be extended significantly to form conditions, which again are only satisfiable by a non-separable penalty, whereby all local minima are smoothed away [28]. Note that while basic ℓ_1-norm minimization also has no local minima, the global minimum need not always correspond with the global solution to (1), unlike when using $g_{\mathrm{SBL}}(\boldsymbol{x})$. As shown in the Appendix, $g_{\mathrm{SBL}}(\boldsymbol{x})$ is a non-decreasing concave function of $|\boldsymbol{x}| \triangleq [|x_1|, \ldots, |x_m|]^T$, and therefore also $\boldsymbol{x}^2 \triangleq [x_1^2, \ldots, x_m^2]^T$, so (perhaps not surprisingly) minimization can be accomplished using either iterative reweighted ℓ_2 or ℓ_1. In this section, we consider the former; the sequel addresses the latter.

There exist multiple ways to handle ℓ_2 reweighting in terms of the non-separable penalty function (19) for the purpose of solving

$$\min_{\boldsymbol{x}} \|\boldsymbol{y} - \Phi\boldsymbol{x}\|_2^2 + \lambda g_{\mathrm{SBL}}(\boldsymbol{x}). \tag{20}$$

Ideally, we would form a quadratic upper bound as follows. Since $g_{\mathrm{SBL}}(\boldsymbol{x})$ is concave and non-decreasing with respect to $\boldsymbol{x}^2$, it can be expressed as

$$g_{\mathrm{SBL}}(\boldsymbol{x}) = \min_{\boldsymbol{z}\geq 0} \boldsymbol{z}^T\boldsymbol{x}^2 - h^*(\boldsymbol{z}) \tag{21}$$

where $h^*(\boldsymbol{z})$ denotes the concave conjugate of $h(\boldsymbol{u}) \triangleq g_{\mathrm{SBL}}(\sqrt{\boldsymbol{u}})$, with $\boldsymbol{u} = \boldsymbol{x}^2 \geq 0$ and $\sqrt{\boldsymbol{u}} \triangleq [\sqrt{u_1}, \ldots, \sqrt{u_m}]^T$. The non-negativity of $\boldsymbol{z}$ follows from the non-decreasing property of $g_{\mathrm{SBL}}(\boldsymbol{x})$ with respect to $\boldsymbol{x}^2$. The concave conjugate is a function that arises from convex analysis and duality theory [1]; in this instance it is expressed via

$$h^*(\boldsymbol{z}) = \min_{\boldsymbol{u}\geq 0} \boldsymbol{z}^T\boldsymbol{u} - h(\boldsymbol{u}) = \min_{\boldsymbol{x}} \boldsymbol{z}^T\boldsymbol{x}^2 - g_{\mathrm{SBL}}(\boldsymbol{x}). \tag{22}$$

If we drop the minimization from (21), we obtain a rigorous quadratic upper bound on $g_{\mathrm{SBL}}(\boldsymbol{x})$. We can then iteratively solve

$$\min_{\boldsymbol{x};\boldsymbol{z}\geq 0} \|\boldsymbol{y} - \Phi\boldsymbol{x}\|_2^2 + \lambda\left[\sum_i z_i x_i^2 - h^*(\boldsymbol{z})\right]. \tag{23}$$

over $\boldsymbol{x}$ and $\boldsymbol{z}$. The $\boldsymbol{x}$ update becomes (3), while the optimal $\boldsymbol{z}$ is given by the gradient of $g_{\mathrm{SBL}}(\boldsymbol{x})$ with respect to $\boldsymbol{x}^2$, or equivalently, the gradient of $h(\boldsymbol{u})$ with respect to $\boldsymbol{u}$ [1]. This gives the weights

$$\begin{aligned} w_i^{(k+1)} \to z_i &= \left.\frac{\partial h(\boldsymbol{u})}{\partial u_i}\right|_{\boldsymbol{u}=\left(\boldsymbol{x}^{(k+1)}\right)^2} \\ &= \left.\frac{\partial g_{\mathrm{SBL}}(\boldsymbol{x})}{\partial x_i^2}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(k+1)}}. \end{aligned} \tag{24}$$

However, this quantity is unfortunately not available in closed form (as far as we know). Alternatively, the use of different sets of upper-bounding auxiliary functions (which are tight in different regions of $\boldsymbol{x}$ space) lead to different choices for $\boldsymbol{w}^{(k+1)}$ with different properties.

One useful variant that reveals a close connection with Chartrand and Yin's method and produces simple, tractable reweighted ℓ_2 updates can be derived as follows. Using the

definition of $g_{\rm SBL}(\boldsymbol{x})$ and standard determinant identities we get

$$\begin{aligned} g_{\rm SBL}(\boldsymbol{x}) &\leq \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \log|\alpha I + \Phi\Gamma\Phi^T| \\ &= \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \log|\Gamma| \\ &\quad + \log|\alpha^{-1}\Phi^T\Phi + \Gamma^{-1}| + n\log\alpha \\ &\leq \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \log|\Gamma| \\ &\quad + \boldsymbol{z}^T\boldsymbol{\gamma}^{-1} - h^*(\boldsymbol{z}) + n\log\alpha \\ &= n\log\alpha - h^*(\boldsymbol{z}) + \sum_i \left(\frac{x_i^2 + z_i}{\gamma_i} + \log\gamma_i\right) \end{aligned} \tag{25}$$

where $h^*(\boldsymbol{z})$ now denotes the concave conjugate of $h(\boldsymbol{a}) \triangleq \log|\alpha^{-1}\Phi^T\Phi + A|$, with $\boldsymbol{a} = \text{diag}[A] = [a_1, \ldots, a_m]^T$ and $A = \Gamma^{-1}$. This conjugate function is computed via

$$h^*(\boldsymbol{z}) = \min_{\boldsymbol{a}\geq 0} \boldsymbol{z}^T\boldsymbol{a} - \log|\alpha^{-1}\Phi^T\Phi + A|. \tag{26}$$

The bound (25) holds for all non-negative vectors $\boldsymbol{z}$ and $\boldsymbol{\gamma}$. We can then perform coordinate descent over

$$\min_{\boldsymbol{x};\boldsymbol{z},\boldsymbol{\gamma}\geq 0} \|\boldsymbol{y} - \Phi\boldsymbol{x}\|_2^2 + \lambda\left[-h^*(\boldsymbol{z}) + \sum_i \left(\frac{x_i^2 + z_i}{\gamma_i} + \log\gamma_i\right)\right]. \tag{27}$$

where irrelevant terms have been dropped. As before, the $\boldsymbol{x}$ update becomes (3). The optimal $\boldsymbol{z}$ is given by

$$\begin{aligned} z_i &= \frac{\partial \log|\alpha^{-1}\Phi^T\Phi + A|}{\partial a_i} \\ &= [(\alpha^{-1}\Phi^T\Phi + \Gamma^{-1})^{-1}]_{ii} \\ &= \gamma_i - \gamma_i^2\boldsymbol{\phi}_i^T(\alpha I + \Phi\Gamma\Phi^T)^{-1}\boldsymbol{\phi}_i, \quad \forall i \end{aligned} \tag{28}$$

which can be stably computed even with $\alpha \to 0$ using the Moore–Penrose pseudo-inverse. Finally, since the optimal γ_i for fixed $\boldsymbol{z}$ and $\boldsymbol{x}$ satisfies $\gamma_i = x_i^2 + z_i, \forall i$, the new weight update becomes

$$\begin{aligned} w_i^{(k+1)} \to \gamma_i^{-1} = \Bigg[&\left(x_i^{(k+1)}\right)^2 + \left(w_i^{(k)}\right)^{-1} \\ &- \left(w_i^{(k)}\right)^{-2}\boldsymbol{\phi}_i^T(\alpha I + \Phi\tilde{W}^{(k)}\Phi^T)^{-1}\boldsymbol{\phi}_i\Bigg]^{-1} \end{aligned} \tag{29}$$

which can be computed in $O(n^2m)$, the same expense as each solution to (3). These updates are guaranteed to reduce or leave unchanged (20) at each iteration.[4] Note that since each weight update is dependent on previous weight updates, it is implicitly dependent on previous values of $\boldsymbol{x}$, unlike in the separable cases above.

[4]They do not however satisfy all of the technical conditions required to ensure global convergence to a local minima (for the same reason that reweighted ℓ_2 is not globally convergent for minimizing the ℓ_1 norm), although in practice we have not observed any problem.

The form of (29) is very similar to the one used by Chartrand and Yin. Basically, if we allow for a separate ϵ_i for each coefficient x_i, then the update (7) is equivalent to the selection

$$\epsilon_i^{(k+1)} \to \left(w_i^{(k)}\right)^{-1} - \left(w_i^{(k)}\right)^{-2}\boldsymbol{\phi}_i^T(\alpha I + \Phi\tilde{W}^{(k)}\Phi^T)^{-1}\boldsymbol{\phi}_i. \tag{30}$$

Moreover, the implicit auxiliary function from (9) being minimized by Chartrand and Yin's method has the exact same form as (25); with the latter, coefficients that are interrelated by a non-separable penalty term are effectively decoupled when conditioned on the auxiliary variables $\boldsymbol{z}$ and $\boldsymbol{\gamma}$. Also recall that one outstanding issue with Chartrand and Yin's approach is the optimal schedule for adjusting $\epsilon^{(k)}$, which could be application-dependent and potentially sensitive.[5] So in this regard, (30) can be viewed as a principled way of selecting ϵ so as to avoid, where possible, convergence to local minima. In preliminary experiments, this method performs as well or better than the heuristic ϵ-selection strategy from [4] (see Sections V-A and V-C).

B. ℓ_1 Reweighting Applied to $g_{\rm SBL}(\boldsymbol{x})$

As mentioned previously, $g_{\rm SBL}(\boldsymbol{x})$ is a non-decreasing, concave function of $|\boldsymbol{x}|$ (see Appendix for details), a desirable property of sparsity-promoting penalties. Importantly, as a direct consequence of this concavity, (20) can potentially be optimized using a reweighted ℓ_1 algorithm (in an analogous fashion to the reweighted ℓ_2 case) using

$$w_i^{(k+1)} \to \left.\frac{\partial g_{\rm SBL}(\boldsymbol{x})}{\partial|x_i|}\right|_{\boldsymbol{x}=\boldsymbol{x}^{(k+1)}}. \tag{31}$$

Like the ℓ_2 case, this quantity is not available in closed form (except for the special case where $\alpha \to 0$). However, as shown in the Appendix it can be iteratively computed by executing the following.

1) Initialization: set $\boldsymbol{w}^{(k+1)} \to \boldsymbol{w}^{(k)}$, the kth vector of weights,
2) Repeat until convergence[6]

$$w_i^{(k+1)} \to \left[\boldsymbol{\phi}_i^T\left(\alpha I + \Phi\tilde{W}^{(k+1)}\tilde{X}^{(k+1)}\Phi^T\right)^{-1}\boldsymbol{\phi}_i\right]^{\frac{1}{2}} \tag{32}$$

where $\tilde{W}^{(k+1)} \triangleq \text{diag}[\boldsymbol{w}^{(k+1)}]^{-1}$ as before and $\tilde{X}^{(k+1)} \triangleq \text{diag}[|\boldsymbol{x}^{(k+1)}|]$. Note that cost function descent is guaranteed with only a single iteration, so we need not execute (32) until convergence. In fact, it can be shown that a more rudimentary form of reweighted ℓ_1 applied to this model in [27] amounts to performing exactly one such iteration, and satisfies all the conditions required for guaranteed convergence (by virtue of the Global Convergence Theorem) to a stationary point

[5]Note that with $\alpha \to 0$ (which seems from empirical results to be the optimal choice), computing $\epsilon_i^{(k+1)}$ via (30) is guaranteed to satisfy $\epsilon_i^{(k+1)} \to 0$ at any stationary point. For other values of α, the $\epsilon_i^{(k+1)}$ associated with nonzero coefficients many potentially remain nonzero, although this poses no problem with respect to sparsity or convergence issues, etc.

[6]Just to clarify, the index k specifies the outer-loop iteration number from (4); for simplicity we have omitted a second index to specify the inner-loop iterations considered here for updating $\boldsymbol{w}^{(k+1)}$.

of (20) (see [27, Theorem 1]). Note however that repeated execution of (32) is very cheap computationally since it scales as $O(nm\|\boldsymbol{x}^{(k+1)}\|_0)$, and is substantially less intensive than the subsequent ℓ_1 step given by (4).[7]

From a theoretical standpoint, ℓ_1 reweighting applied to $g_{\rm SBL}(\boldsymbol{x})$ is guaranteed to aid performance in the sense described by the following two results, which apply to the canonical sparse recovery problem (1). Before proceeding, we define $\mathrm{spark}(\Phi)$ as the smallest number of linearly dependent columns in Φ [8]. It follows then that $2 \leq \mathrm{spark}(\Phi) \leq n+1$.

Theorem 1: Assume $\lambda \rightarrow 0, \alpha \rightarrow 0$. When applying iterative reweighted ℓ_1 using (32) and $w_i^{(1)} < \infty, \forall i$, the solution sparsity satisfies $\|\boldsymbol{x}^{(k+1)}\|_0 \leq \|\boldsymbol{x}^{(k)}\|_0$ (i.e., continued iteration can never do worse).

Theorem 2: Assume $\lambda \rightarrow 0$,$\alpha \rightarrow 0$, and that $\mathrm{spark}(\Phi) = n+1$ and consider any instance where standard ℓ_1 minimization fails to find some $\boldsymbol{x}^*$ drawn from support set $\mathcal{S}$ with cardinality $|\mathcal{S}| < (n+1)/(2)$. Then there exists a set of signals $\boldsymbol{y}$ (with nonzero measure) generated from $\mathcal{S}$ such that non-separable reweighted ℓ_1, with $\boldsymbol{w}^{(k+1)}$ updated using (32), always succeeds but standard ℓ_1 always fails.

Note that Theorem 2 does not in any way indicate what is the best non-separable reweighting scheme in practice (for example, in our limited experience with empirical simulations, the selection $\alpha \rightarrow 0$ is not necessarily always optimal). However, it does suggest that reweighting with non-convex, non-separable penalties is potentially very effective, motivating other selections as discussed next. Taken together, Theorems 1 and 2 challenge the prevailing reliance on strictly convex cost functions, since they ensure that we can never do worse than the minimum ℓ_1-norm solution (which uses the tightest convex approximation to the ℓ_0 norm), and that there will always be cases where improvement over this solution is obtained.

Before proceeding, it is worth relating Theorem 2 with Proposition 5 from Davies and Gribonval [5], where it is shown that for any sparsity level, there will always exist cases (albeit of measure zero) where, if standard ℓ_1 minimization fails, any *admissible* ℓ_1 reweighting strategy will also fail. In this context a reweighting scheme is said to be admissible if: 1) $w_i^{(1)} = 1$ for all i and, 2) there exists a $w_{\max}^{(k)} < \infty$ such that for all k and i, $w_i^{(k)} \leq w_{\max}^{(k)}$ and if $x_i^{(k)} = 0$, then $w_i^{(k)} = w_{\max}^{(k)}$.

Interestingly, the non-separable reweighting from (32) does *not* satisfy this definition despite its effectiveness in practice (see Sections V-B and V-C below). It fails for two reasons: First, $w_{\max}^{(k)} \rightarrow \infty$ as $\alpha \rightarrow 0$, and second, the condition $x_i^{(k)} = 0$ does not ensure that $w_i^{(k)} = w_{\max}^{(k)}$. Yet it is this failure to always assign the largest weight to zero-valued coefficients that helps non-separable methods avoid bad local minima (see Section III-C for more details), and so we suggest modified versions of admissibility that accommodate a wider class of useful non-separable algorithms.

[7]While a similar inner-loop iterative procedure could potentially be adopted to estimate (24), this is not practical for two reasons. First, because sparse solutions are not obtained after each reweighted ℓ_2 iteration, the per-iteration cost of the inner-loop would be more expensive. Second, because many more outer-loop reweighted ℓ_2 iterations are required (each of which is relatively cheap on its own), the total cost of the inner-loops will be substantially higher.

One alternative definition, which is consistent with convergence considerations and the motivation first used to inspire many iterative reweighting algorithms, is simply to require that an admissible ℓ_1 reweighting scheme is one such that $g(\boldsymbol{x}^{(k+1)}) \leq g(\boldsymbol{x}^{(k)})$ for all k, where $g(\cdot)$ is some non-decreasing, concave function of $|\boldsymbol{x}|$.[8] Note that this function need not be available in closed form to satisfy this definition; practical, admissible algorithms can nonetheless be obtained even when $g(\cdot)$ can only be computed numerically. Section III-C gives one such example. Additionally, it can be shown via simple counterexamples that Proposition 5 from [5] explicitly does not hold given this updated notion of admissibility.

In summary, we would argue that a broader conception of reweighting strategies can potentially be advantageous for avoiding local minima and finding maximally sparse solutions. Moreover, we stress the contrasting nature of Davies and Gribonval's result versus our Theorem 2. The former demonstrates that on a set of measure zero in $\boldsymbol{x}$ space, a particular class of reweighting schemes will not improve upon basic ℓ_1 minimization, while the latter specifies that on a different set of nonzero measure, some non-separable reweighting will always do better.

C. Bottom-Up Construction of Non-Separable Penalty Using Reweighted ℓ_1

In the previous section, we described what amounts to a top-down formulation of a non-separable penalty function that emerges from a particular hierarchical Bayesian model. Based on the insights gleaned from this procedure (and its distinction from separable penalties), it is possible to stipulate alternative penalty functions from the bottom up by creating plausible, non-separable reweighting schemes. The following is one such possibility.

Assume for simplicity that $\lambda \rightarrow 0$. The Achilles heel of standard, separable penalties is that if we want to retain a global minimum similar to that of (1), we require a highly concave penalty on each x_i [28]. However, this implies that almost all *basic feasible solutions* (BFS) to $\boldsymbol{y} = \Phi\boldsymbol{x}$, defined as a solution with $\|\boldsymbol{x}\|_0 \leq n$, will form local minima of the penalty function constrained to the feasible region. This is a very undesirable property since there are on the order of $\binom{m}{n}$ BFS with $\|\boldsymbol{x}\|_0 = n$, which is equal to the signal dimension and not very sparse. We would really like to find *degenerate* BFS, where $\|\boldsymbol{x}\|_0$ is strictly less than n. Such solutions are exceedingly rare and difficult to find. Consequently we would like to utilize a non-separable, yet highly concave penalty that explicitly favors degenerate BFS. We can accomplish this by constructing a reweighting scheme designed to avoid non-degenerate BFS whenever possible.

Now consider the covariance-like quantity $\alpha I + \Phi(\tilde{X}^{(k+1)})^2\Phi^T$, where α may be small, and then construct weights using the projection of each basis vector $\boldsymbol{\phi}_i$ as defined via

$$w_i^{(k+1)} \rightarrow \boldsymbol{\phi}_i^T \left(\alpha I + \Phi(\tilde{X}^{(k+1)})^2\Phi^T\right)^{-1} \boldsymbol{\phi}_i. \qquad (33)$$

[8]A analogous admissibility condition for the reweighted ℓ_2 case would require concavity with respect to $\boldsymbol{x}^2$.

Ideally, if at iteration $k+1$ we are at a bad or non-degenerate BFS, we do not want the newly computed $w_i^{(k+1)}$ to favor the present position at the next iteration of (4) by assigning overly large weights to the zero-valued x_i. In such a situation, the factor $\Phi(\hat{X}^{(k+1)})^2\Phi^T$ in (33) will be full rank and so all weights will be relatively modest sized. In contrast, if a rare, degenerate BFS is found, then $\Phi(\hat{X}^{(k+1)})^2\Phi^T$ will no longer be full rank, and the weights associated with zero-valued coefficients will be set to large values, meaning this solution will be favored in the next iteration.

In some sense, the distinction between (33) and its separable counterparts, such as the method of Candès *et al.* which uses (12), can be summarized as follows: the separable methods assign the largest weight *whenever* the associated coefficient goes to zero; with (33) the largest weight is only assigned when the associated coefficient goes to zero *and* $\|\boldsymbol{x}^{(k+1)}\|_0 < n$, which differs significantly from Davies and Gribonval's notion of an admissible weighting scheme.

The reweighting option (33), which bears some resemblance to (32), also has some very desirable properties beyond the intuitive justification given above. First, since we are utilizing (33) in the context of reweighted ℓ_1 minimization, it would be productive to know what cost function, if any, we are minimizing when we compute each iteration. Using the fundamental theorem of calculus for line integrals (or the gradient theorem), it follows that the bottom-up (BU) penalty function associated with (33) is

$$g_{\rm BU}(\boldsymbol{x}) \triangleq \int_0^1 \mathrm{trace}[\hat{X}\Phi^T(\alpha I + \Phi(\nu\hat{X})^2\Phi^T)^{-1}\Phi]d\nu. \tag{34}$$

Moreover, because each weight w_i is a non-increasing function of each $x_j, \forall j$, from Kachurovskii's theorem [21] it directly follows that (34) is concave and non-decreasing in $|\boldsymbol{x}|$, and thus naturally promotes sparsity. Additionally, for α sufficiently small, it can be shown that the global minimum of (34) on the constraint $\boldsymbol{y} = \Phi\boldsymbol{x}$ must occur at a degenerate BFS (Theorem 1 from above also holds when using (33); Theorem 2 may as well, although we have not formally shown this). And finally, regarding implementational issues and interpretability, (33) avoids any recursive weight assignments or inner-loop optimization as when using (32). Empirical experiments using this method are presented in Sections V-B and V-C.

IV. Extensions

One of the motivating factors for using iterative reweighted optimization, especially the ℓ_1 variant, is that it is often very easy to incorporate alternative data-fit terms, constraints, and sparsity penalties. This section addresses two useful examples.

Non-Negative Sparse Coding: Numerous applications require sparse solutions where all coefficients x_i are constrained to be non-negative [2]. By adding the contraint $\boldsymbol{x} \geq 0$ to (4) at each iteration, we can easily compute such solutions using $g_{\rm SBL}(\boldsymbol{x})$, $g_{\rm BU}(\boldsymbol{x})$, or any other appropriate penalty function. Note that in the original SBL formulation, this is not a possibility since the integrals required to compute the associated cost function and update rules no longer have closed-form expressions.

Group Feature Selection: Another common generalization is to seek sparsity at the level of groups of features, e.g., the group Lasso [30]. The simultaneous sparse approximation problem [19], [25] is a particularly useful adaptation of this idea relevant to compressive sensing [26], manifold learning [22], and neuroimaging [29]. In this situation, we are presented with r signals $Y \triangleq [\boldsymbol{y}_{\cdot 1}, \boldsymbol{y}_{\cdot 2}, \ldots, \boldsymbol{y}_{\cdot r}]$ that we assume were produced by coefficient vectors $X \triangleq [\boldsymbol{x}_{\cdot 1}, \boldsymbol{x}_{\cdot 2}, \ldots, \boldsymbol{x}_{\cdot r}]$ characterized by the same sparsity profile or support, meaning that the coefficient matrix X is row sparse. Note that to facilitate later analysis, we adopt the notation that $\boldsymbol{x}_{\cdot j}$ represents the jth column of X while $\boldsymbol{x}_{i\cdot}$ represents the ith row of X.

As an extension of the ℓ_0 norm to the simultaneous approximation problem, we define

$$d(X) \triangleq \sum_{i=1}^{m} \mathcal{I}[\|\boldsymbol{x}_{i\cdot}\| > 0] \tag{35}$$

where $\mathcal{I}[\|\boldsymbol{x}\| > 0] = 1$ if $\|\boldsymbol{x}\| > 0$ and zero otherwise, and $\|\boldsymbol{x}\|$ is an arbitrary vector norm. $d(X)$ penalizes the number of rows in X that are not equal to zero; for nonzero rows there is no additional penalty for large magnitudes. Also, for the column vector $\boldsymbol{x}$, it is immediately apparent that $d(\boldsymbol{x}) = \|\boldsymbol{x}\|_0$. Given this definition, the sparse recovery problems (1) and (2) become

$$\min_X d(X), \quad \text{s.t. } Y = \Phi X \tag{36}$$

and

$$\min_X \|Y - \Phi X\|_{\mathcal{F}}^2 + \lambda d(X), \quad \lambda > 0. \tag{37}$$

As before, the combinatorial nature of each optimization problem renders them intractable and so approximate procedures are required. All of the algorithms discussed herein can naturally be expanded to this domain essentially by substituting the scalar coefficient magnitudes from a given iteration $|x_i^{(k)}|$ with some row-vector penalty, such as a norm. For the iterative reweighted ℓ_2 methods to work seamlessly, we require the use of $\|\boldsymbol{x}_{i\cdot}\|_2$ and everything proceeds exactly as before. In contrast, the iterative reweighted ℓ_1 situation is both more flexible and somewhat more complex. If we utilize $\|\boldsymbol{x}_{i\cdot}\|_2$, then the coefficient matrix update analogous to (4) requires the solution of the more complicated weighted second-order cone (SOC) program

$$X^{(k+1)} \rightarrow \arg\min_X \|Y - \Phi X\|_{\mathcal{F}}^2 + \lambda \sum_i w_i^{(k)} \|\boldsymbol{x}_{i\cdot}\|_2. \tag{38}$$

Other selections such as the ℓ_∞ norm are possible as well, providing added generality to this approach.

Both separable and non-separable methods lend themselves equally well to the simultaneous sparse approximation problem. Preliminary results in Section V-C however, indicate that non-separable algorithms can significantly widen their performance advantage over their separable counterparts in this domain.

V. Empirical Results

This section contains a few brief experiments involving the various reweighting schemes discussed previously. First, we in-

clude a comparison of ℓ_2 approaches followed by an ℓ_1 example involving non-negative sparse coding. We then conclude with simulations involving the simultaneous sparse approximation problem (group sparsity), where we compare ℓ_1 and ℓ_2 algorithms head-to-head. In all cases, we use dashed lines to denote the performance of separable algorithms, while solid lines represent the non-separable ones.

A. Iterative Reweighted ℓ_2 Examples

Monte-Carlo simulations were conducted similar to those performed in [4] and [6] allowing us to compare the separable method of Chartrand and Yin with the non-separable SBL update (29) using $\alpha \to 0$. As discussed above, these methods differ only in the effective choice of the ϵ parameter. We also include results from the related method in Daubechies *et al.* [6] using $p = 1$, which gives us the basis pursuit or Lasso (minimum ℓ_1 norm) solution, and $p = 0.6$ which works well in conjunction with the proscribed ϵ update based on the simulations from [6]. Note that the optimal value of p and ϵ for sparse recovery purposes can be interdependent and [6] reports poor results with p much smaller than 0.6 when using their ϵ update. Additionally, there is an additional parameter K associated with Daubechies *et al.*'s ϵ update that must be set; we used the heuristic taken from the authors' Matlab code.[9]

The experimental particulars are as follows. First, a random, overcomplete 50×250 dictionary Φ is created with independent and identically distributed (i.i.d.) unit Gaussian elements and ℓ_2 normalized columns. Next, sparse coefficient vectors $\boldsymbol{x}^*$ are randomly generated with the number of nonzero entries varied to create different test conditions. Nonzero amplitudes are drawn i.i.d. from one of two experiment-dependent distributions. Signals are then computed as $\boldsymbol{y} = \Phi \boldsymbol{x}^*$. Each algorithm is presented with $\boldsymbol{y}$ and Φ and attempts to estimate $\boldsymbol{x}^*$ using an initial weighting of $w_i^{(1)} = 1, \forall i$. In all cases, we ran 1000 independent trials and compared the number of times each algorithm failed to recover $\boldsymbol{x}^*$. Under the specified conditions for the generation of Φ and $\boldsymbol{y}$, all other feasible solutions $\boldsymbol{x}$ are almost surely less sparse than $\boldsymbol{x}^*$, so our synthetically generated coefficients will almost surely be maximally sparse.

Fig. 1 displays results where the nonzero elements in $\boldsymbol{x}^*$ were drawn with unit magnitudes. The performance of four algorithms is shown: the three separable methods discussed above and the non-separable update given by (29) and referred to as SBL-ℓ_2. For algorithms with non-convex underlying sparsity penalties, unit magnitude coefficients can be much more troublesome than other distributions because local minima may become more pronounced or numerous [28]. In contrast, the performance will be independent of the nonzero coefficient magnitudes when minimizing the ℓ_1 norm (i.e., the $p = 1$ case) [15], so we expect this situation to be most advantageous to the ℓ_1-norm solution relative to the others. Nevertheless, from the figure we observe that the non-separable reweighting still performs best; out of the remaining separable examples, the $p = 1$ case is only slightly superior.

Regarding computational complexity, the individual updates associated with the various reweighted ℓ_2 algorithms have roughly the same expense. Consequently, it is the number of iterations that can potentially distinguish the effective complexity of the different methods. Table I compares the number of iterations required by each algorithm when $\|\boldsymbol{x}^*\|_0 = 12$ such that the maximum change of any estimated coefficient was less than 10^{-6}. We also capped the maximum number of iterations at 1000. Results are partitioned into cases where the algorithms were successful, meaning the correct $\boldsymbol{x}^*$ was obtained, versus failure cases where a suboptimal solution was obtained. This distinction was made because generally convergence is much faster when the correct solution has been found, so results could be heavily biased by the success rate of the algorithm.

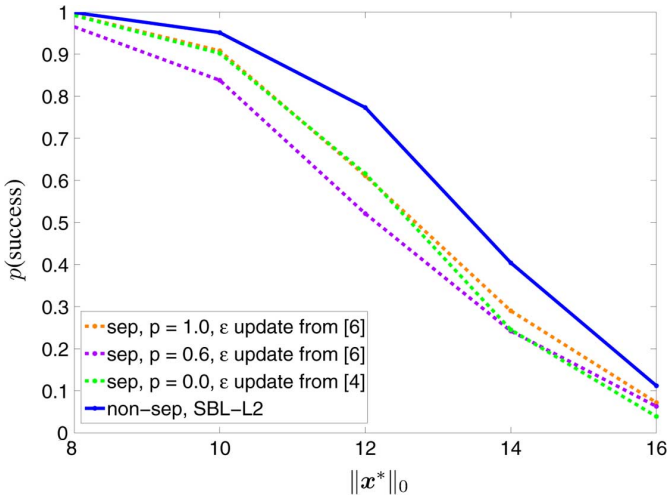

Fig. 1. Iterative reweighted ℓ_2 example using *unit magnitude* nonzero coefficients. Probability of success recovering sparse coefficients for different sparsity values, i.e., $\|\boldsymbol{x}^*\|_0$.

Table I reveals that in the successful cases, the separable update from [6], with $p = 0.6$ is the most efficient, but this is largely because this algorithm uses knowledge of the true sparsity to shrink ϵ quickly, an advantage not given to the other methods. It is also much faster than the $p = 1.0$ version because the convergence rate (once ϵ becomes small enough), is roughly proportional to $2 - p$, as shown in [20]. In the unsuccessful cases, the separable updates from [6] cannot ever capitalize on the known sparsity level and they do not fully converge even after 1000 iterations.

Fig. 2 reproduces the same experiment with nonzero elements in $\boldsymbol{x}^*$ now drawn from a unit Gaussian distribution. The performance of the $p = 1$ separable algorithm is unchanged as expected; however, the others all improve significantly, especially the non-separable update and Chartrand and Yin's method. Note however that only the non-separable method is parameter-free, in the sense that ϵ is set automatically, and we assume $\alpha \to 0$ (the effectiveness of this value holds up in a wide range of simulations (not shown) where dictionary type and dimensions, as well as signal distributions, are all varied). Regardless, we did find that adding an additional decaying regularization term to (29), updated using a simple heuristic like Chartrand and Yin's method, could improve performance still further. Of course presumably all of these methods could potentially be enhanced through additional modifications and tuning (e.g., a simple hybrid scheme is suggested in [6] that involves reducing p after a "burn-in" period that improves recovery probabilities marginally); however, we save thorough evaluation of such extensions to future exploration.

[9]http://www.ricam.oeaw.ac.at/people/page/fornasier/menu3.html.

TABLE I
AVERAGE NUMBER OF ITERATIONS USED BY EACH ALGORITHM FOR THE $\|\boldsymbol{x}^*\|_0 = 12$ CASE IN FIG. 1. EACH METHOD WAS TERMINATED WHEN THE MAXIMUM CHANGE OF ANY COEFFICIENT WAS LESS THAN 10^{-6} OR THE MAXIMUM NUMBER OF ITERATIONS, SET TO 1000, WAS REACHED

Algorithm	Separable, ϵ update from [6], $p = 1.0$	Separable, ϵ update from [6], $p = 0.6$	Separable, ϵ update from [4], $p = 0.0$	Non-Separable, SBL-ℓ_2
Average Iterations (Successful Cases)	253.2	14.0	67.7	86.3
Average Iterations (Failure Cases)	1000	1000	355.4	515.7

TABLE II
AVERAGE NUMBER OF ITERATIONS USED BY EACH ALGORITHM FOR THE $\|\boldsymbol{x}^*\|_0 = 16$ CASE IN FIG. 2. EACH METHOD WAS TERMINATED WHEN THE MAXIMUM CHANGE OF ANY COEFFICIENT WAS LESS THAN 10^{-6} OR THE MAXIMUM NUMBER OF ITERATIONS, SET TO 1000, WAS REACHED

Algorithm	Separable, ϵ update from [6], $p = 1.0$	Separable, ϵ update from [6], $p = 0.6$	Separable, ϵ update from [4], $p = 0.0$	Non-Separable, SBL-ℓ_2
Average Iterations (Successful Cases)	400.5	49.2	97.8	107.6
Average Iterations (Failure Cases)	1000	1000	361.3	495.8

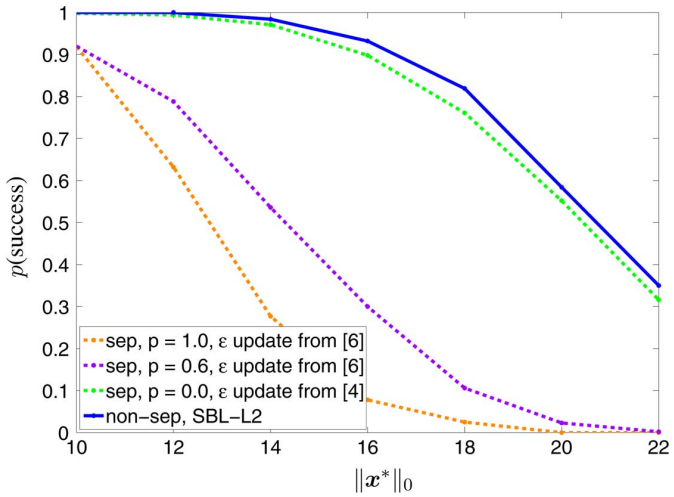

Fig. 2. Iterative reweighted ℓ_2 example using *Gaussian distributed* nonzero coefficients. Probability of success recovering sparse coefficients for different sparsity values, i.e., $\|\boldsymbol{x}^*\|_0$.

We also compared the average number of iterations required for the case where $\|\boldsymbol{x}^*\|_0 = 16$. Results, which are similar to those from Table I with respect to relative efficiency, are shown in Table II.

B. Non-Negative Sparse Coding Example via Reweighted ℓ_1 Minimization

To test reweighted ℓ_1 minimization applied to various sparsity penalties, we performed simulations similar to those in [3]. Each trial consisted of generating a 100×256 dictionary Φ with i.i.d. Gaussian entries and a sparse vector $\boldsymbol{x}^*$ with 60 nonzero, non-negative (truncated Gaussian) coefficients. The signal is computed using $\boldsymbol{y} = \Phi\boldsymbol{x}^*$ as before. We then attempted to recover $\boldsymbol{x}^*$ by applying non-negative ℓ_1 reweighting strategies with three different penalty functions: 1) $g_{\rm SBL}(\boldsymbol{x})$ implemented using (32), referred to as SBL-ℓ_1, 2) $g_{\rm BU}(\boldsymbol{x})$, and 3) $g(\boldsymbol{x}) = \sum_i \log(|x_i| + \epsilon)$, the separable method of Candès *et al.*, which represents the current state-of-the-art in reweighted ℓ_1 algorithms. In each case, the algorithm's tuning parameter (either α or ϵ) was chosen via coarse cross-validation (see below). Additionally, since we are working with a noise-free signal, we assume $\lambda \to 0$ and so the requisite coefficient update (4) with $x_i \geq 0$ reduces to the standard linear program

$$\boldsymbol{x}^{(k+1)} \to \arg\min_{\boldsymbol{x}} \sum_i w_i^{(k)} x_i, \quad \text{s.t. } \boldsymbol{y} = \Phi\boldsymbol{x}, \quad x_i \geq 0, \quad \forall i. \tag{39}$$

Given $w_i^{(1)} = 1, \forall i$ for each algorithm, the first iteration amounts to the non-negative minimum ℓ_1-norm solution. Average results from 1000 random trials are displayed in Fig. 3, which plots the empirical probability of success in recovering $\boldsymbol{x}^*$ versus the iteration number. We observe that standard non-negative ℓ_1 never succeeds (see first iteration results); however, with only a few reweighted iterations drastic improvement is possible, especially for the non-separable algorithms. None of the algorithms improved appreciably after ten iterations (not shown). These results show both the efficacy of non-separable reweighting and the ability to handle constraints on $\boldsymbol{x}$.

To roughly assess how performance varies with the tuning parameters, α for the non-separable methods, ϵ for Candès *et al.*, we repeated the above experiment using a 50×100 dictionary and $\|\boldsymbol{x}^*\|_0 = 30$, while α, ϵ were varied from 10^{-4} to 10^4. Empirical performance (after ten reweighted ℓ_1 iterations) as a function of α, ϵ is displayed in Fig. 4 below. Two things are worth pointing out with respect to these results. First, the performance overall of the non-separable approaches is superior to the method of Candès *et al.* Second, as $\alpha, \epsilon \to \infty$, the performance of all algorithms converges to that of the standard (non-negative)

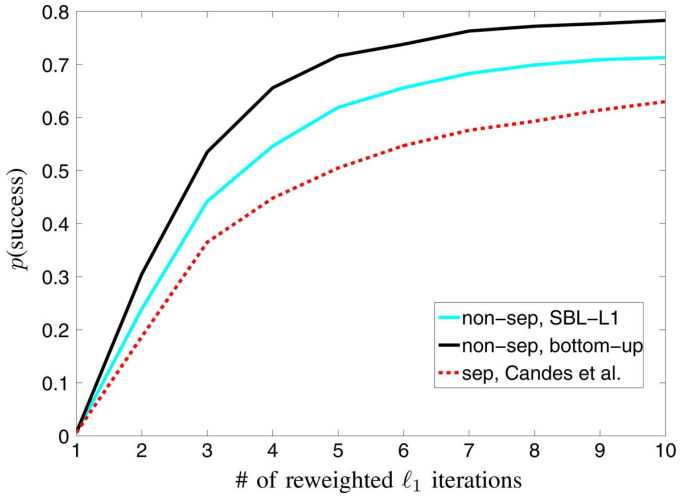

Fig. 3. Non-negative sparse coding example. Probability of success recovering sparse, non-negative coefficients as the number of reweighted ℓ_1 iterations is increased.

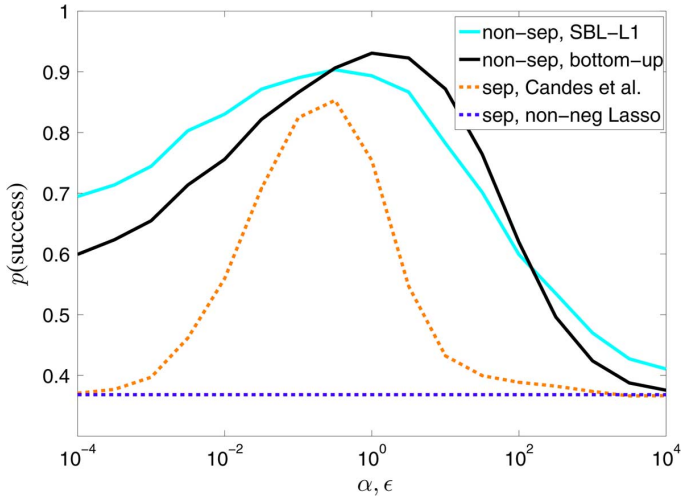

Fig. 4. Probability of success recovering sparse, non-negative coefficients as a function of α, ϵ using ten reweighted ℓ_1 iterations. The non-negative Lasso has no such parameter, so its performance is flat.

minimum ℓ_1-norm solution (i.e., the non-negative Lasso) as expected by theory (assuming the dictionary columns have unit ℓ_2 norm). So asymptotically to the right there is very little distinction. In contrast, when $\alpha, \epsilon \to 0$, the story is very different. The performance of Candès *et al.*'s method again theoretically converges to the minimum ℓ_1-norm solution. This is because after one iteration the resulting solution will necessarily be a local minima, so further iterations will offer no improvement. However, with the non-separable algorithms this is not the case; multiple iterations retain the possibility of improvement, consistent with the analysis of Section III. Hence, their performance with $\alpha \to 0$ is well above the ℓ_1-norm solution.

As mentioned previously, $\alpha \to 0$ seems to always be optimal for the reweighted ℓ_2 version of SBL whereas this is clearly not the case in the reweighted ℓ_1 example in Fig. 4. To provide one possible explanation as to why this may be reasonable, it is helpful to consider how $g_{\rm SBL}(\boldsymbol{x})$ is affected by α. When α small, the penalty $g_{\rm SBL}(\boldsymbol{x})$ is more highly concave (favoring sparsity) and the global minimum will always match that of the minimum ℓ_0-norm solution. When using reweighted ℓ_2 this is desirable since the algorithm takes relatively slow progressive steps, avoiding most local minima and converging to the ideal global solution. In contrast, reweighted ℓ_1 converges much more quickly and may be more prone to local minima when α is too small. When α is a bit larger, reweighted ℓ_1 performance improves as local minima are smoothed; however, the convergence rate of reweighted ℓ_2 drops and the global minima may no longer always be optimal.

C. Application of Reweighting Schemes to the Simultaneous Sparse Approximation Problem

For this experiment we used a randomly generated 50×100 dictionary for each trial with i.i.d. Gaussian entries normalized as above, and created five coefficient vectors $X^* = [\boldsymbol{x}^*_{\cdot 1}, \ldots, \boldsymbol{x}^*_{\cdot 5}]$ with matching sparsity profile and i.i.d. Gaussian nonzero coefficients. We then generate the signal matrix $Y = \Phi X^*$ and attempt to learn X^* using various group-level reweighting schemes. In this experiment, we varied the row sparsity of X^* from $d(X^*) = 30$ to $d(X^*) = 40$; in general, the more nonzero rows, the harder the recovery problem becomes.

A total of seven algorithms modified to the simultaneous sparse approximation problem were tested using an ℓ_2-norm penalty on each coefficient row. This requires that the reweighted ℓ_1 variants compute the SOC program (38) as discussed in Section IV. The seven algorithms used were: 1) SBL-ℓ_2, 2) SBL-ℓ_1, 3) the non-separable bottom up approach, 4) Candès *et al.*'s method minimized with reweighted ℓ_2 via (15), 5) the original ℓ_1 Candès *et al.* method, 6) Chartrand and Yin's method, and finally 7), the standard group Lasso [30] (equivalent to a single iteration of any of the other reweighted ℓ_1 algorithms). Note that for method 6), unlike the others, there is no reweighted ℓ_1 analog because the underlying penalty function is not concave with respect to each $|x_i|$ or $\|\boldsymbol{x}_{i\cdot}\|_2$, it is only concave with respect to x_i^2 or $\|\boldsymbol{x}_{i\cdot}\|_2^2$. For methods 1)–3) we simply used $\alpha \to 0$ for direct comparison. For 4) and 5) we chose a fixed ϵ using cross-validation. Finally, for 6) we used the heuristic described in [4]. Note that in all cases, we actually used the weighted norm $(1)/(\sqrt{r})\|\boldsymbol{x}_{i\cdot}\|_2$ to maintain some consistency with the $r = 1$ case in setting tuning parameters.

Results are presented in Fig. 5. The three non-separable algorithms display the best performance, with the ℓ_2 version overwhelmingly so. We also observe that the ℓ_2 variant of Candès *et al.*'s method that we derived significantly outperforms its ℓ_1 counterpart from [3]. Again though, we stress that the ℓ_1 updates are generally much more flexible regarding additional constraints applied either to the data fit term or the sparsity penalty and may benefit from efficient convex programming toolboxes. We also note that the popular group Lasso is decidedly worse than all of the other algorithms, implying that the requirement of a strictly convex cost function can be very detrimental to performance.

VI. Conclusion

In this paper, we have briefly explored various iterative reweighting schemes for solving sparse linear inverse problems, elaborating on a distinction between separable and non-separable weighting functions and sparsity penalties. Although a large number of separable algorithms have been proposed in the

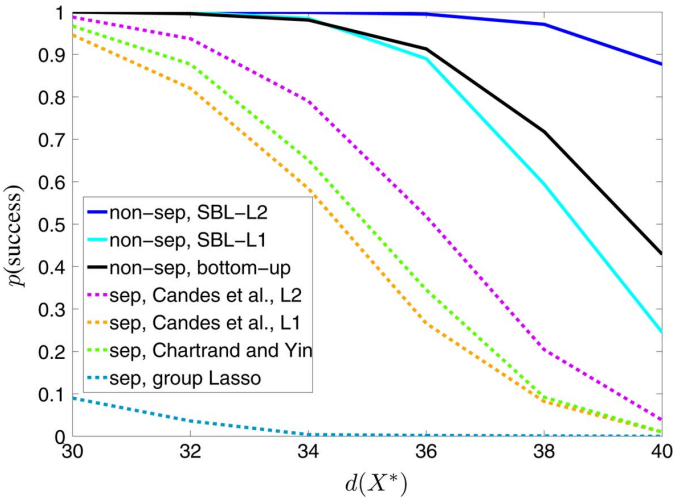

Fig. 5. Iterative reweighted results using five simultaneous signal vectors. Probability of success recovering sparse coefficients for different row sparsity values, i.e., $d(X^*)$.

literature, the non-separable case is relatively uncommon and, on the surface, may appear much more difficult to work with. However, iterative reweighted ℓ_1 and ℓ_2 approaches provide a convenient means of decoupling coefficients via auxiliary variables leading to efficient updates that can potentially be related to existing separable schemes. In general, a variety of different algorithms are possible by forming different upper-bounding auxiliary functions. While the non-separable algorithms we have derived show considerable promise, we envision that superior strategies and interesting extensions are very possible. In practice, we have successfully applied this methodology to large neuroimaging data sets obtaining significant improvements over existing convex approaches such as the group Lasso [29].

APPENDIX

Concavity of $g_{\rm SBL}(\boldsymbol{x})$ and Derivation of Weight Updates (32): Because $\log|\alpha I+\Phi\Gamma\Phi^T|$ is concave and non-decreasing with respect to $\boldsymbol{\gamma}\geq 0$, we can express it as

$$\log|\alpha I+\Phi\Gamma\Phi^T| = \min_{\boldsymbol{z}\geq 0} \boldsymbol{z}^T\boldsymbol{\gamma} - h^*(\boldsymbol{z}) \tag{40}$$

where $h^*(\boldsymbol{z})$ is defined as the concave conjugate [1] of $h(\boldsymbol{\gamma}) \triangleq \log|\alpha I+\Phi\Gamma\Phi^T|$ given by

$$h^*(\boldsymbol{z}) = \min_{\boldsymbol{\gamma}\geq 0} \boldsymbol{z}^T\boldsymbol{\gamma} - \log|\alpha I+\Phi\Gamma\Phi^T|. \tag{41}$$

We can then express $g_{\rm SBL}(\boldsymbol{x})$ via

$$\begin{aligned} g_{\rm SBL}(\boldsymbol{x}) &= \min_{\boldsymbol{\gamma}\geq 0} \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \log|\alpha I+\Phi\Gamma\Phi^T| \\ &= \min_{\boldsymbol{\gamma},\boldsymbol{z}\geq 0} \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \boldsymbol{z}^T\boldsymbol{\gamma} - h^*(\boldsymbol{z}) \\ &= \min_{\boldsymbol{\gamma},\boldsymbol{z}\geq 0} \sum_i \left(\frac{x_i^2}{\gamma_i} + z_i\gamma_i\right) - h^*(\boldsymbol{z}). \end{aligned} \tag{42}$$

Minimizing over $\boldsymbol{\gamma}$ for fixed $\boldsymbol{x}$ and $\boldsymbol{z}$, we get

$$\gamma_i = z_i^{-1/2}|x_i|, \quad \forall i. \tag{43}$$

Substituting this expression into (42) gives the representation

$$\begin{aligned} g_{\rm SBL}(\boldsymbol{x}) &= \min_{\boldsymbol{z}\geq 0} \sum_i \left(\frac{x_i^2}{z_i^{-1/2}|x_i|} + z_i z_i^{-1/2}|x_i|\right) - h^*(\boldsymbol{z}) \\ &= \min_{\boldsymbol{z}\geq 0} \sum_i 2z_i^{1/2}|x_i| - h^*(\boldsymbol{z}) \end{aligned} \tag{44}$$

which implies that $g_{\rm SBL}(\boldsymbol{x})$ can be expressed as a minimum of upper-bounding hyperplanes with respect to $|\boldsymbol{x}|$, and thus must be concave and non-decreasing in $|\boldsymbol{x}|$ since $\boldsymbol{z}\geq 0$ [1]. We also observe that for fixed $\boldsymbol{z}$, solving (20) is a weighted ℓ_1 minimization problem.

To derive the weight update (32), we only need the optimal value of each z_i, which from basic convex analysis will satisfy

$$z_i^{1/2} = \frac{\partial g_{\rm SBL}(\boldsymbol{x})}{2\partial|x_i|}. \tag{45}$$

Since this quantity is not available in closed form, we can instead iteratively minimize (42) over γ and $\boldsymbol{z}$. We start by initializing $z_i^{1/2} \rightarrow w_i^{(k)}, \forall i$ and then minimize over $\boldsymbol{\gamma}$ using (43). We then compute the optimal $\boldsymbol{z}$ for fixed $\boldsymbol{\gamma}$, which can be done analytically using

$$\begin{aligned} \boldsymbol{z} &= \nabla_{\boldsymbol{\gamma}} \log|\alpha I+\Phi\Gamma\Phi^T| \\ &= \mathrm{diag}[\Phi^T(\alpha I+\Phi\Gamma\Phi^T)^{-1}\Phi]. \end{aligned} \tag{46}$$

This result follows from basic principles of convex analysis. By substituting (43) into (46) and setting $w_i^{(k+1)} \rightarrow z_i^{1/2}$, we obtain the weight update (32).

We now demonstrate that this iterative process will converge monotonically to a solution which globally minimizes the bound from (42) with respect to $\boldsymbol{\gamma}$ and $\boldsymbol{z}$, and therefore gives a solution for $\boldsymbol{z}$ which satisfies (45). Monotonic convergence to some local minimum (or saddle point) is guaranteed using the same analysis from [27] referred to in Section III-B. It only remains to be shown that no suboptimal local minima or saddle points exist.

For this purpose, it is sufficient to show that $\rho(\boldsymbol{\gamma}) \triangleq \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \log|\alpha I+\Phi\Gamma\Phi^T|$ (i.e., we have reinserted the optimal $\boldsymbol{z}$ as a function of $\boldsymbol{\gamma}$ into the bound) is unimodal with respect to $\boldsymbol{\gamma}$ with no suboptimal local minima or saddle points. To accomplish this, we use

$$\begin{aligned} \rho(\boldsymbol{\gamma}) &\equiv \boldsymbol{x}^T\Gamma^{-1}\boldsymbol{x} + \log|\Gamma| \\ &\quad + \log|\alpha^{-1}\Phi^T\Phi + \Gamma^{-1}| \\ &= \sum_i \left[x_i^2\exp(-\theta_i) + \theta_i\right] \\ &\quad + \log|\alpha^{-1}\Phi^T\Phi + \mathrm{diag}[\exp(-\boldsymbol{\theta})]| \end{aligned} \tag{47}$$

where $\theta_i \triangleq \log\gamma_i$ for all i, $\boldsymbol{\theta} \triangleq [\theta_1,\ldots,\theta_m]^T$, and the $\exp[\cdot]$ operator is understood to apply element-wise. The first term above is obviously convex in $\boldsymbol{\theta}$. Taking the Hessian of the last term with respect to $\boldsymbol{\theta}$ reveals that it is positive semi-definite, and therefore convex in $\boldsymbol{\theta}$ as well. Consequently, the function $\rho(\exp[\boldsymbol{\theta}])$ is convex with respect to $\boldsymbol{\theta}$. Since $\exp[\cdot]$ is a smooth monotonically increasing function, this implies that $\rho(\boldsymbol{\gamma})$ will be unimodal (although not necessarily convex) in $\boldsymbol{\gamma}$. ■

Proof of Theorem 1: Before we begin, we should point out that for $\alpha \rightarrow 0$, the weight update (32) is still well-specified regardless of the value of the diagonal matrix $\tilde{W}^{(k+1)}\tilde{X}^{(k+1)}$. If

$\boldsymbol{\phi}_i$ is not in the span of $\Phi\tilde{W}^{(k+1)}\tilde{X}^{(k+1)}\Phi^T$, then $w_i^{(k+1)} \to \infty$ and the corresponding coefficient x_i can be set to zero for all future iterations. Otherwise, $w_i^{(k+1)}$ can be computed efficiently using the Moore–Penrose pseudoinverse and will be strictly finite.

For simplicity we will now assume that $\mathrm{spark}(\Phi) = n+1$, which is equivalent to requiring that each subset of n columns of Φ forms a basis in $\mathbb{R}^n$. The extension to the more general case is discussed below. From basic linear programming [14], at any iteration the coefficients will satisfy $\|\boldsymbol{x}^{(k)}\|_0 \leq n$ for arbitrary weights $\boldsymbol{w}^{(k-1)}$. Given our simplifying assumptions, there exists only two possibilities. If $\|\boldsymbol{x}^{(k)}\|_0 = n$, then we will automatically satisfy $\|\boldsymbol{x}^{(k+1)}\|_0 \leq \|\boldsymbol{x}^{(k)}\|_0$ at the next iteration regardless of $\tilde{W}^{(k)}$. In contrast, if $\|\boldsymbol{x}^{(k)}\|_0 < n$, then $\mathrm{rank}[\tilde{W}^{(k)}] \leq \|\boldsymbol{x}^{(k)}\|_0$ for all evaluations of (32) with $\alpha \to 0$, enforcing $\|\boldsymbol{x}^{(k+1)}\|_0 \leq \|\boldsymbol{x}^{(k)}\|_0$.

If $\mathrm{spark}(\Phi) < n+1$, then the globally minimizing weighted ℓ_1 solution may not be unique. In this situation, there will still always be a global minimizer such that $\|\boldsymbol{x}^{(k)}\|_0 \leq n$; however, there may be others with $\|\boldsymbol{x}^{(k)}\|_0 > n$ forming a convex solution set. To satisfy the theorem then, one would need to use an ℓ_1 solver that always returns the sparsest element of this minimum ℓ_1-norm solution set. However, we should point out that this is a very minor contingency in practice, in part because it has been well-established that essentially all useful, random matrices will satisfy $\mathrm{spark}(\Phi) = n+1$ with overwhelming probability. ■

Proof of Theorem 2: For a fixed dictionary Φ and coefficient vector $\boldsymbol{x}^*$, we are assuming that $\|\boldsymbol{x}^*\|_0 < (n+1)/(2)$. Now consider a second coefficient vector $\boldsymbol{x}'$ with support and sign pattern equal to $\boldsymbol{x}^*$ and define $x'_{(i)}$ as the ith largest coefficient magnitude of $\boldsymbol{x}'$. Then there exists a set of $\|\boldsymbol{x}^*\|_0 - 1$ scaling constants $\nu_i \in (0, 1]$ (i.e., strictly greater than zero) such that, for any signal $\boldsymbol{y}$ generated via $\boldsymbol{y} = \Phi\boldsymbol{x}'$ and $x'_{(i+1)} \leq \nu_i x'_{(i)}$, $i = 1, \ldots, \|\boldsymbol{x}^*\|_0 - 1$, the minimization problem

$$\hat{\boldsymbol{x}} \triangleq \arg\min_{\boldsymbol{x}} g_{\mathrm{SBL}}(\boldsymbol{x}), \quad \text{s.t. } \Phi\boldsymbol{x}' = \Phi\boldsymbol{x}, \alpha \to 0 \tag{48}$$

is unimodal and has a unique minimizing stationary point which satisfies $\hat{\boldsymbol{x}} = \boldsymbol{x}'$. This result follows from [28].

Note that (48) is equivalent to (20) with $\lambda \to 0$, so the reweighted non-separable update (32) can be applied. Furthermore, based on the global convergence of these updates discussed above, the sequence of estimates are guaranteed to satisfy $\boldsymbol{x}^{(k)} \to \hat{\boldsymbol{x}} = \boldsymbol{x}'$. So we will necessarily learn the generative $\boldsymbol{x}'$.

Let $\boldsymbol{x}^{\ell_1} \triangleq \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1$, subject to $\Phi\boldsymbol{x}^* = \Phi\boldsymbol{x}$. By assumption we know that $\boldsymbol{x}^{\ell_1} \neq \boldsymbol{x}^*$. Moreover, we can conclude using [15, Theorem 6] that if $\boldsymbol{x}^{\ell_1}$ fails for some $\boldsymbol{x}^*$, it will fail for any other $\boldsymbol{x}$ with matching support and sign pattern; it will therefore fail for any $\boldsymbol{x}'$ as defined above.[10] Finally, by construction, the set of feasible $\boldsymbol{x}'$ will have nonzero measure over the support $\mathcal{S}$ since each ν_i is strictly nonzero. Note also that this result can likely be extended to the case where $\mathrm{spark}(\Phi) < n+1$ and to any $\boldsymbol{x}^*$ that satisfies $\|\boldsymbol{x}^*\|_0 < \mathrm{spark}(\Phi) - 1$. The more specific case addressed above was only assumed to allow direct application of [15, Theorem 6]. ■

[10]Actually, [15, Theorem 6] shows that if $\boldsymbol{x}^{\ell_1}$ succeeds for a given support and sign pattern of $\boldsymbol{x}^*$, then it will succeed regardless of the magnitudes of $\boldsymbol{x}^*$. However, this naturally implies that if $\boldsymbol{x}^{\ell_1}$ fails, it will fail for all magnitudes. To see this suppose that the negative statement were not true and that $\boldsymbol{x}^{\ell_1}$ fails for $\boldsymbol{x}^*$ but then succeeds for some other $\boldsymbol{x}'$ with the same sparsity profile and sign pattern. This would contradict the original theorem because success with $\boldsymbol{x}'$ would then imply success recovering $\boldsymbol{x}^*$.

REFERENCES

[1] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[2] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "A non-negative and sparse enough solution of an underdetermined linear system of equations is unique," *IEEE Trans. Inf. Theory*, vol. 54, no. 11, pp. 4813–4820, Nov. 2008.

[3] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5, pp. 877–905, Dec. 2008.

[4] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. Int. Conf. Accoust., Speech, Signal Process.*, 2008, pp. 3869–3872.

[5] M. Davies and R. Gribonval, "Restricted isometry constants where ℓ_p Sparse recovery can fail for $0 < p \leq 1$," Tech. Rep. IRISA, no. 1899, Jul. 2008.

[6] I. Daubechies, R. DeVore, M. Fornasier, and S. Gunturk, "Iteratively re-weighted least squares minimization for sparse recovery," *Commun. Pure Appl. Math.*, vol. 63, no. 1, pp. 1–38, 2010.

[7] D. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," Stanford Univ. Tech. Rep., Stanford, CA, Sep. 2004.

[8] D. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proc. National Acad. Sci.*, vol. 100, no. 5, pp. 2197–2202, Mar. 2003.

[9] M. Fazel, H. Hindi, and S. Boyd, "Log-Det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices," in *Proc. Amer. Control Conf.*, Jun. 2003, vol. 3, pp. 2156–2162.

[10] M. Figueiredo, "Adaptive sparseness for supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 9, pp. 1150–1159, Sep. 2003.

[11] M. Figueiredo, J. Bioucas-Dias, and R. Nowak, "Majorization-minimization algorithms for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2980–2991, Dec. 2007.

[12] J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1341–1344, Jun. 2004.

[13] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Trans. Inf. Theory*, vol. 49, no. 12, pp. 3320–3325, Dec. 2003.

[14] G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. Reading, MA: Addison-Wesley, 1984.

[15] D. Malioutov, M. Çetin, and A. Willsky, "Optimal sparse representations in general overcomplete bases," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2004, vol. 2, pp. II-793–II-796.

[16] R. Neal, *Bayesian Learning for Neural Networks*. New York: Springer-Verlag, 1996.

[17] J. Palmer, D. Wipf, K. Kreutz-Delgado, and B. Rao, "Variational EM algorithms for non-gaussian latent variable models," *Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 1059–1066, 2006.

[18] B. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado, "Subset selection in noise based on diversity measure minimization," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 760–770, Mar. 2003.

[19] B. D. Rao and K. Kreutz-Delgado, "Basis selection in the presence of noise," in *Proc. 32nd Asilomar Conf. Signals, Syst. Comput.*, Nov. 1998, vol. 1, pp. 752–756.

[20] B. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 187–200, Jan. 1999.

[21] R. Showalter, "Monotone operators in Banach space and nonlinear partial differential equations," Mathematical Surveys and Monographs vol. 49, 1997, AMS, Providence, RI.

[22] J. Silva, J. Marques, and J. Lemos, "Selecting landmark points for sparse manifold learning," *Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 1241–1248, 2006.

[23] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[24] J. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[25] J. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Process.*, vol. 86, pp. 589–602, Apr. 2006.

[26] M. Wakin, M. Duarte, S. Sarvotham, D. Baron, and R. Baraniuk, "Recovery of jointly sparse signals from a few random projections," *Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 1433–1440, 2006.
[27] D. Wipf and S. Nagarajan, "A new view of automatic relevance determination," *Adv. Neural Inf. Process. Syst.*, vol. 20, 2008.
[28] D. Wipf and S. Nagarajan, "Latent variable Bayesian models for promoting sparsity," , submitted for publication, 2009.
[29] D. Wipf, J. Owen, H. Attias, K. Sekihara, and S. Nagarajan, "Robust Bayesian estimation of the location, orientation, and timecourse of mutliple correlated neural sources using MEG," *NeuroImage*, 2009, to be published.
[30] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. R. Statist. Soc. B*, vol. 68, pp. 49–67, 2006.
[31] W. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1969.

David Wipf received the B.S. degree in electrical engineering from the University of Virginia, Charlottesville, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, San Diego.

Currently, he is an NIH Postdoctoral Fellow in the Biomagnetic Imaging Lab, University of California, San Francisco. His research involves the development and analysis of Bayesian learning algorithms for functional brain imaging and sparse coding.

Srikantan Nagarajan received the M.S. and Ph.D. degrees in biomedical engineering from Case Western Reserve University, Cleveland, OH.

He received a postdoctoral fellowship at the Keck Center for Integrative Neuroscience at the University of California, San Francisco (UCSF). Currently, he is a Professor in the Department of Radiology and Biomedical Imaging, UCSF, and a faculty member in the UCSF/UC Berkeley Joint Graduate Program in Bioengineering. His research interests, in the area of Neural Engineering and machine learning, are to better understand neural mechanisms of sensorimotor learning and speech motor control, to develop algorithms for improved functional brain imaging and biomedical signal processing, and to refine clinical applications of functional brain imaging in patients with stroke, schizophrenia, autism, brain tumors, epilepsy, focal hand dystonia, and Parkinson's disease.