SIAM J. Imaging Sciences
Vol. 6, No. 4, pp. 2010–2046

A Method for Finding Structured Sparse Solutions to Nonnegative Least Squares Problems with Applications*

Ernie Esser†, Yifei Lou†, and Jack Xin†

Abstract. Unmixing problems in many areas such as hyperspectral imaging and differential optical absorption spectroscopy (DOAS) often require finding sparse nonnegative linear combinations of dictionary elements that match observed data. We show how aspects of these problems, such as misalignment of DOAS references and uncertainty in hyperspectral endmembers, can be modeled by expanding the dictionary with grouped elements and imposing a structured sparsity assumption that the combinations within each group should be sparse or even 1-sparse. If the dictionary is highly coherent, it is difficult to obtain good solutions using convex or greedy methods, such as nonnegative least squares (NNLS) or orthogonal matching pursuit. We use penalties related to the Hoyer measure, which is the ratio of the l_1 and l_2 norms, as sparsity penalties to be added to the objective in NNLS-type models. For solving the resulting nonconvex models, we propose a scaled gradient projection algorithm that requires solving a sequence of strongly convex quadratic programs. We discuss its close connections to convex splitting methods and difference of convex programming. We also present promising numerical results for DOAS analysis and hyperspectral unmixing problems.

Key words. unmixing, nonnegative least squares, basis pursuit, structured sparsity, scaled gradient projection, difference of convex programming, hyperspectral imaging, differential optical absorption spectroscopy

AMS subject classifications. 90C55, 90C90, 65K10, 49N45

DOI. 10.1137/13090540X

1. Introduction.

A general unmixing problem is to estimate the quantities or concentrations of the individual components of some observed mixture. Often a linear mixture model is assumed [39]. In this case the observed mixture b is modeled as a linear combination of references for each component known to possibly be in the mixture. If we put these references in the columns of a dictionary matrix A, then the mixing model is simply $Ax = b$. Physical constraints often mean that x should be nonnegative, and, depending on the application, we may also be able to make sparsity assumptions about the unknown coefficients x. This can be posed as a basis pursuit problem where we are interested in finding a sparse and perhaps also nonnegative linear combination of dictionary elements that match observed data. This is a very well studied problem. Some standard convex models are nonnegative least squares (NNLS) [42, 53], i.e.,

$$\min_{x\geq 0} \frac{1}{2}\|Ax-b\|^2, \tag{1.1}$$

and methods based on l_1 minimization [15, 59, 63].

*Received by the editors January 9, 2013; accepted for publication (in revised form) July 1, 2013; published electronically October 22, 2013. This work was partially supported by NSF grants DMS-0911277, DMS-0928427, and DMS-1222507.
http://www.siam.org/journals/siims/6-4/90540.html

†Department of Mathematics, UC Irvine, Irvine, CA 92697 (eesser@uci.edu, ylou1@uci.edu, jxin@math.uci.edu).

In this paper we are interested in how to deal with uncertainty in the dictionary. The case when the dictionary is unknown is dealt with in sparse coding and nonnegative matrix factorization (NMF) problems [49, 46, 30, 43, 4, 18], which require learning both the dictionary and a sparse representation of the data. We are, however, interested in the case where we know the dictionary but are uncertain about each element. One example we will study in this paper is differential optical absorption spectroscopy (DOAS) analysis [50], for which we know the reference spectra but are uncertain about how to align them with the data because of wavelength misalignment. Another example we will consider is hyperspectral unmixing [8, 27, 29]. Multiple reference spectral signatures, or endmembers, may have been measured for the same material, and they may all be slightly different if they were measured under different conditions. We may not know ahead of time how to choose the one that is most consistent with the measured data. Spectral variability of endmembers has been introduced in previous works, for example, in [55, 17, 32, 67, 16], and includes considering noise in the endmembers and representing endmembers as random vectors. However, we may not always have a good general model for endmember variability. For the DOAS example, we do have a good model for the unknown misalignment [50], but even so, incorporating it may significantly complicate the overall model. Therefore, for both examples, instead of attempting to model the uncertainty, we propose to expand the dictionary to include a representative group of possible elements for each uncertain element as was done in [44].

The grouped structure of the expanded dictionary is known by construction, and this allows us to make additional structured sparsity assumptions about the corresponding coefficients. In particular, the coefficients should be extremely sparse within each group of representative elements, and in many cases we would like them to be at most 1-sparse. We will refer to this as intragroup sparsity. If we expected sparsity of the coefficients for the unexpanded dictionary, then this will carry over to an intergroup sparsity assumption about the coefficients for the expanded dictionary. By intergroup sparsity we mean that with the coefficients split into groups, the number of groups containing nonzero elements should also be sparse. Examples of existing structured sparsity models include group lasso [23, 64, 47, 51] and exclusive lasso [68]. More general structured sparsity strategies that include applying sparsity penalties separately to possibly overlapping subsets of variables can be found in [36, 37, 3, 33].

The expanded dictionary we consider is usually an underdetermined matrix with the property of being highly coherent because the added columns tend to be similar to each other. This makes it very challenging to find good sparse representations of the data using standard convex minimization and greedy optimization methods. If A satisfies certain properties related to its columns not being too coherent [11], then sufficiently sparse nonnegative solutions are unique and can therefore be found by solving the convex NNLS problem. These assumptions are usually not satisfied for our expanded dictionaries, and while NNLS may still be useful as an initialization, it does not by itself produce sufficiently sparse solutions. Similarly, our expanded dictionaries usually do not satisfy the incoherence assumptions required for l_1 minimization or for greedy methods like orthogonal matching pursuit (OMP) to recover the l_0 sparse solution [60, 12]. However, with an unexpanded dictionary having relatively few columns, these techniques can be effectively used for sparse hyperspectral unmixing [35].

The coherence of our expanded dictionary means that we need to use different tools to find good solutions that satisfy our sparsity assumptions. We would like to use a variational

approach as similar as possible to the NNLS model that enforces the additional sparsity while still allowing all the groups to collaborate. We propose adding nonconvex sparsity penalties to the NNLS objective function (1.1). We can apply these penalties separately to each group of coefficients to enforce intragroup sparsity, and we can simultaneously apply them to the vector of all coefficients to enforce additional intergroup sparsity. From a modeling perspective, the ideal sparsity penalty is l_0. There is a very interesting recent work that deals directly with l_0 constraints and penalties via a quadratic penalty approach [45]. If the variational model is going to be nonconvex, we prefer to work with a differentiable objective when possible. We therefore explore the effectiveness of sparsity penalties based on the Hoyer measure [31, 34], which is essentially the ratio of l_1 and l_2 norms. In previous works, this has been successfully used to model sparsity in NMF and blind deconvolution applications [31, 40, 38]. We also consider the difference of l_1 and l_2 norms. By the relationship $\|x\|_1 - \|x\|_2 = \|x\|_2(\frac{\|x\|_1}{\|x\|_2} - 1)$, we see that while the ratio of norms is constant in radial directions, the difference increases moving away from the origin except along the axes. Since the Hoyer measure is twice differentiable on the nonnegative orthant away from the origin, it can be locally expressed as a difference of convex functions, and convex splitting or difference of convex (DC) methods [57] can be used to find a local minimum of the nonconvex problem. Some care must be taken, however, to deal with the Hoyer measure's poor behavior near the origin. It is even easier to apply DC methods when using $l_1 - l_2$ as a penalty, since this is already a difference of convex functions and is well defined at the origin.

The paper is organized as follows. In section 2 we define the general model, describe the dictionary structure, and show how to use both the ratio and the difference of l_1 and l_2 norms to model our intra- and intergroup sparsity assumptions. Section 3 derives a method for solving the general model, discusses connections to existing methods, and includes convergence analysis. In section 4 we discuss specific problem formulations for several examples related to DOAS analysis and hyperspectral unmixing. Numerical experiments for comparing methods and applications to example problems are presented in section 5.

2. Problem. For the nonnegative linear mixing model $Ax = b$, let $b \in \mathbb{R}^W$, $A \in R^{W\times N}$, and $x \in \mathbb{R}^N$ with $x \geq 0$. Let the dictionary A have l_2 normalized columns and consist of M groups, each with m_j elements. We can write $A = \begin{bmatrix} A_1 & \cdots & A_M \end{bmatrix}$ and $x = \begin{bmatrix} x_1 & \cdots & x_M \end{bmatrix}^T$, where each $x_j \in \mathbb{R}^{m_j}$ and $N = \sum_{j=1}^M m_j$. The general NNLS problem with sparsity constraints that we will consider is

$$\min_{x\geq 0} F(x) := \frac{1}{2}\|Ax - b\|^2 + R(x), \tag{2.1}$$

where

$$R(x) = \sum_{j=1}^{M} \gamma_j R_j(x_j) + \gamma_0 R_0(x). \tag{2.2}$$

The functions R_j represent the intrasparsity penalties applied to each group of coefficients x_j, $j = 1, \ldots, M$, and R_0 is the intersparsity penalty applied to x. If F is differentiable, then a

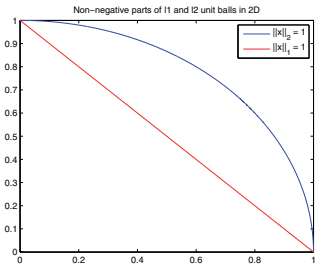

Figure 1. *l_1 and l_2 unit balls.*

necessary condition for x^* to be a local minimum is given by

$$(y - x^*)^T \nabla F(x^*) \geq 0 \qquad \forall y \geq 0. \tag{2.3}$$

For the applications we will consider, we want to constrain each vector x_j to be at most 1-sparse, which is to say that we want $\|x_j\|_0 \leq 1$. To accomplish this through the model (2.1), we will need to choose the parameters γ_j to be sufficiently large.

The sparsity penalties R_j and R_0 will either be the ratios of l_1 and l_2 norms defined by

$$H_j(x_j) = \gamma_j \frac{\|x_j\|_1}{\|x_j\|_2} \qquad \text{and} \qquad H_0(x) = \gamma_0 \frac{\|x\|_1}{\|x\|_2}, \tag{2.4}$$

or they will be the differences defined by

$$S_j(x_j) = \gamma_j(\|x_j\|_1 - \|x_j\|_2) \qquad \text{and} \qquad S_0(x) = \gamma_0(\|x\|_1 - \|x\|_2). \tag{2.5}$$

A geometric intuition for why minimizing $\frac{\|x\|_1}{\|x\|_2}$ promotes sparsity of x is that since it is constant in radial directions, minimizing it tries to reduce $\|x\|_1$ without changing $\|x\|_2$. As seen in Figure 1, sparser vectors have a smaller l_1 norm on the l_2 sphere.

Neither H_j nor S_j is differentiable at zero, and H_j is not even continuous there. Figure 2 shows a visualization of both penalties in two dimensions. To obtain a differentiable F, we can smooth the sparsity penalties by replacing the l_2 norm with the Huber function, defined by the infimal convolution

$$\phi(x, \epsilon) = \inf_y \|y\|_2 + \frac{1}{2\epsilon}\|y - x\|^2 = \begin{cases} \frac{\|x\|_2^2}{2\epsilon} & \text{if } \|x\|_2 \leq \epsilon, \\ \|x\|_2 - \frac{\epsilon}{2} & \text{otherwise.} \end{cases} \tag{2.6}$$

In this way we can define differentiable versions of sparsity penalties H and S by

$$H_j^{\epsilon_j}(x_j) = \gamma_j \frac{\|x_j\|_1}{\phi(x_j, \epsilon_j) + \frac{\epsilon_j}{2}}, \qquad H_0^{\epsilon}(x) = \gamma_0 \frac{\|x\|_1}{\phi(x, \epsilon_0) + \frac{\epsilon_0}{2}}, \tag{2.7}$$

$$S_j^{\epsilon}(x_j) = \gamma_j(\|x_j\|_1 - \phi(x_j, \epsilon_j)), \qquad S_0^{\epsilon}(x) = \gamma_0(\|x\|_1 - \phi(x, \epsilon_0)). \tag{2.8}$$

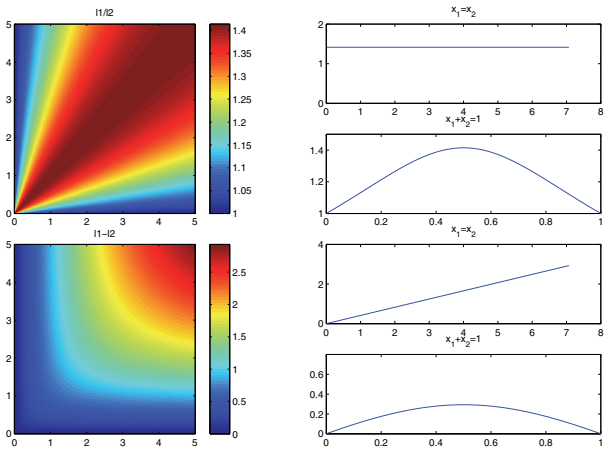

Figure 2. *Visualization of l_1/l_2 and $l_1 - l_2$ penalties.*

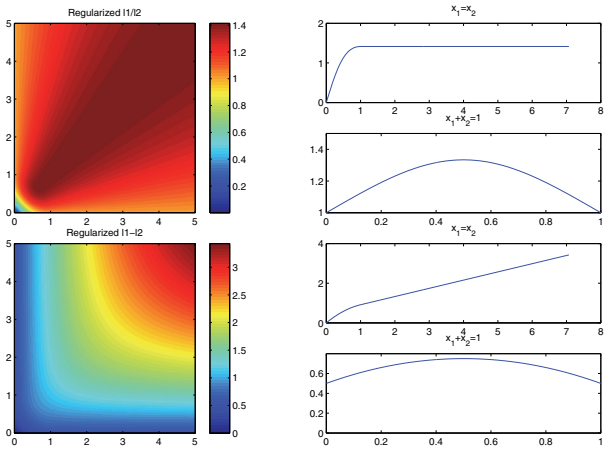

Figure 3. *Visualization of regularized l_1/l_2 and $l_1 - l_2$ penalties.*

These smoothed sparsity penalties are shown in Figure 3. The regularized penalties behave more like l_1 near the origin and should tend to shrink x_j that have small l_2 norms.

An alternate strategy for obtaining a differentiable objective that doesn't require smoothing the sparsity penalties is to add M additional dummy variables and modify the convex constraint set. Let $d \in \mathbb{R}^M$, $d \geq 0$, denote a vector of dummy variables. Consider applying R_j to vectors $\begin{bmatrix} x_j \\ d_j \end{bmatrix}$ instead of to x_j. Then if we add the constraints $\|x_j\|_1 + d_j \geq \epsilon_j$, we are assured that $R_j(x_j, d_j)$ will be applied only to nonzero vectors, even though x_j is still allowed to be zero. Moreover, by requiring that $\sum_j \frac{d_j}{\epsilon_j} \leq M - r$, we can ensure that at least r of the vectors x_j have one or more nonzero elements. In particular, this prevents x from being zero, so $R_0(x)$ is well defined as well.

The dummy variable strategy is our preferred approach for using the l_1/l_2 penalty. The high variability of the regularized version near the origin creates numerical difficulties. Either it needs a lot of smoothing, which makes it behave too much like l_1, or its steepness near the origin makes it harder numerically to avoid getting stuck in bad local minima. For the $l_1 - l_2$ penalty, the regularized approach is our preferred strategy because it is simpler and not much regularization is required. Smoothing also makes this penalty behave more like l_1 near the origin, but a small shrinkage effect there may in fact be useful, especially for promoting intergroup sparsity. These two main problem formulations are summarized below as Problems 1 and 2, respectively.

Problem 1.

$$\min_{x,d} F_H(x,d) := \frac{1}{2}\|Ax-b\|^2 + \sum_{j=1}^{M} \gamma_j H_j(x_j,d_j) + \gamma_0 H_0(x)$$

$$\text{such that} \quad x > 0, \quad d > 0, \quad \sum_{j=1}^{M} \frac{d_j}{\epsilon_j} \le M - r \quad \text{and} \quad \|x_j\|_1 + d_j \ge \epsilon_j,\ j = 1,\dots,M.$$

Problem 2.

$$\min_{x\ge 0} F_S(x) := \frac{1}{2}\|Ax-b\|^2 + \sum_{j=1}^{M} \gamma_j S_j^{\epsilon}(x_j) + \gamma_0 S_0^{\epsilon}(x).$$

3. Algorithm. Both Problems 1 and 2 can be written abstractly as

$$\min_{x\in X} F(x) := \frac{1}{2}\|Ax-b\|^2 + R(x), \tag{3.1}$$

where X is a convex set. Problem 2 is already of this form with $X = \{x \in \mathbb{R}^N : x \ge 0\}$. Problem 1 is also of this form, with $X = \{x \in \mathbb{R}^N, d \in \mathbb{R}^M : x > 0, d > 0, \|x_j\|_1 + d_j \ge \epsilon_j, \sum_j \frac{d_j}{\epsilon_j} \le M - r\}$. Note that the objective function of Problem 1 can also be written as in (3.1) if we redefine x_j as $\begin{bmatrix} x_j \\ d_j \end{bmatrix}$ and consider an expanded vector of coefficients $x \in \mathbb{R}^{N+M}$ that includes the M dummy variables, d. The data fidelity term can still be written as $\frac{1}{2}\|Ax-b\|^2$ if columns of zeros are inserted into A at the indices corresponding to the dummy variables. In this section, we will describe algorithms and convergence analysis for solving (3.1) under either of two sets of assumptions.

Assumption 1.

- *X is a convex set.*
- *$R(x) \in \mathcal{C}^2(X,\mathbb{R})$, and the eigenvalues of $\nabla^2 R(x)$ are bounded on X.*
- *F is coercive on X in the sense that for any $x^0 \in X$, $\{x \in X : F(x) \le F(x^0)\}$ is a bounded set. In particular, F is bounded below.*

Assumption 2.

- *$R(x)$ is concave and differentiable on X.*
- *The same assumptions on X and F as in Assumption 1 hold.*

Problem 1 satisfies Assumption 1, and Problem 2 satisfies Assumption 2. We will first consider the case of Assumption 1.

Our approach for solving (3.1) was originally motivated by a convex splitting technique from [20, 61] that is a semi-implicit method for solving $\frac{dx}{dt} = -\nabla F(x)$, $x(0) = x^0$, when F can be split into a sum of convex and concave functions $F^C(x) + F^E(x)$, both in $\mathcal{C}^2(\mathbb{R}^N, \mathbb{R})$. Let $\lambda_{F^E}^{\max}$ be an upper bound on the eigenvalues of $\nabla^2 F^E$, and let $\lambda_F^{\min}$ be a lower bound on the eigenvalues of $\nabla^2 F$. Under the assumption that $\lambda_{F^E}^{\max} \leq \frac{1}{2}\lambda_F^{\min}$, it can be shown that the update defined by

$$x^{n+1} = x^n - \Delta t(\nabla F^C(x^{n+1}) + \nabla F^E(x^n)) \tag{3.2}$$

doesn't increase F for any time step $\Delta t > 0$. This can be seen by using second order Taylor expansions to derive the estimate

$$F(x^{n+1}) - F(x^n) \leq \left(\lambda_{F^E}^{\max} - \frac{1}{2}\lambda_F^{\min} - \frac{1}{\Delta t}\right)\|x^{n+1} - x^n\|^2. \tag{3.3}$$

This convex splitting approach has been shown to be an efficient method that is much faster than gradient descent for solving phase-field models such as the Cahn–Hilliard equation, which has been used, for example, to simulate coarsening [61] and for image inpainting [5].

By the assumptions on R, we can achieve a convex-concave splitting, $F = F^C + F^E$, by letting $F^C(x) = \frac{1}{2}\|Ax - b\|^2 + \|x\|_C^2$ and $F^E(x) = R(x) - \|x\|_C^2$ for an appropriately chosen positive definite matrix C. We can also use the fact that $F^C(x)$ is quadratic to improve upon the estimate in (3.3) when bounding $F(x^{n+1}) - F(x^n)$ by a quadratic function of x^{n+1}. Then instead of choosing a time step and updating according to (3.2), we can dispense with the time step interpretation altogether and choose an update that reduces the upper bound on $F(x^{n+1}) - F(x^n)$ as much as possible subject to the constraint. This requires minimizing a strongly convex quadratic function over X.

Proposition 3.1. *Let Assumption 1 hold. Also let $\lambda_R^{\min}$ and $\lambda_R^{\max}$ be lower and upper bounds, respectively, on the eigenvalues of $\nabla^2 R(x)$ for $x \in X$. Then for $x, y \in X$ and for any matrix C,*

$$\begin{aligned} F(y) - F(x) \leq (y-x)^T &\left(\left(\lambda_R^{\max} - \frac{1}{2}\lambda_R^{\min}\right)\mathrm{I} - C\right)(y-x) \\ &+ (y-x)^T\left(\frac{1}{2}A^TA + C\right)(y-x) + (y-x)^T\nabla F(x). \end{aligned} \tag{3.4}$$

Proof. The estimate follows from combining several second order Taylor expansions of F and R with our assumptions. First, expanding F about y and using $h = y - x$ to simplify notation, we get that

$$F(x) = F(y) - h^T\nabla F(y) + \frac{1}{2}h^T\nabla^2 F(y - \alpha_1 h)h$$

for some $\alpha_1 \in (0, 1)$. Substituting F as defined by (3.1), we obtain

$$F(y) - F(x) = h^T(A^TAy - A^Tb + \nabla R(y)) - \frac{1}{2}h^TA^TAh - \frac{1}{2}h^T\nabla^2 R(y - \alpha_1 h)h. \tag{3.5}$$

Similarly, we can compute Taylor expansions of R about both x and y:

$$R(x) = R(y) - h^T \nabla R(y) + \frac{1}{2} h^T \nabla^2 R(y - \alpha_2 h) h.$$

$$R(y) = R(x) + h^T \nabla R(x) + \frac{1}{2} h^T \nabla^2 R(x + \alpha_3 h) h.$$

Again, both α_2 and α_3 are in $(0, 1)$. Adding these expressions implies that

$$h^T(\nabla R(y) - \nabla R(x)) = \frac{1}{2} h^T \nabla^2 R(y - \alpha_2 h) h + \frac{1}{2} h^T \nabla^2 R(x + \alpha_3 h) h.$$

From the assumption that the eigenvalues of $\nabla^2 R$ are bounded above by $\lambda_R^{\max}$ on X,

$$h^T(\nabla R(y) - \nabla R(x)) \leq \lambda_R^{\max} \|h\|^2. \tag{3.6}$$

Adding and subtracting $h^T \nabla R(x)$ and $h^T A^T A x$ to (3.5) yields

$$\begin{aligned} F(y) - F(x) &= h^T A^T A h + h^T (A^T A x - A^T b + \nabla R(x)) + h^T (\nabla R(y) - \nabla R(x)) \\ &\quad - \frac{1}{2} h^T A^T A h - \frac{1}{2} h^T \nabla^2 R(y - \alpha_1 h) h \\ &= \frac{1}{2} h^T A^T A h + h^T \nabla F(x) + h^T (\nabla R(y) - \nabla R(x)) - \frac{1}{2} h^T \nabla^2 R(y - \alpha_1 h) h. \end{aligned}$$

Using (3.6),

$$F(y) - F(x) \leq \frac{1}{2} h^T A^T A h + h^T \nabla F(x) - \frac{1}{2} h^T \nabla^2 R(y - \alpha_1 h) h + \lambda_R^{\max} \|h\|^2.$$

The assumption that the eigenvalues of $\nabla^2 R(x)$ are bounded below by $\lambda_R^{\min}$ on X means that

$$F(y) - F(x) \leq \left(\lambda_R^{\max} - \frac{1}{2} \lambda_R^{\min} \right) \|h\|^2 + \frac{1}{2} h^T A^T A h + h^T \nabla F(x).$$

Since the estimate is unchanged by adding and subtracting $h^T C h$ for any matrix C, the inequality in (3.4) follows directly. ∎

Corollary 3.2. *Let C be symmetric positive definite, and let $\lambda_C^{\min}$ denote the smallest eigenvalue of C. If $\lambda_C^{\min} \geq \lambda_R^{\max} - \frac{1}{2} \lambda_R^{\min}$, then for $x, y \in X$,*

$$F(y) - F(x) \leq (y - x)^T \left(\frac{1}{2} A^T A + C \right) (y - x) + (y - x)^T \nabla F(x).$$

A natural strategy for solving (3.1) is then to iterate

$$x^{n+1} = \arg\min_{x \in X} (x - x^n)^T \left(\frac{1}{2} A^T A + C_n \right) (x - x^n) + (x - x^n)^T \nabla F(x^n) \tag{3.7}$$

for C_n chosen to guarantee a sufficient decrease in F. The method obtained by iterating (3.7) can be viewed as an instance of scaled gradient projection [7, 6, 9], where the orthogonal projection of $x^n - (A^T A + 2C_n)^{-1} \nabla F(x^n)$ onto X is computed in the norm $\|\cdot\|_{A^T A + 2C_n}$. The

approach of decreasing F by minimizing an upper bound coming from an estimate such as (3.4) can be interpreted as majorization-minimization or an optimization transfer strategy of defining and minimizing a surrogate function [41], which is done for related applications in [30, 43]. It can also be interpreted as an example of the concave-convex procedure [54, 65], a special case of DC programming [57].

Choosing C_n in such a way that guarantees $(x^{n+1}-x^n)^T((\lambda_R^{\max}-\frac{1}{2}\lambda_R^{\min})\mathrm{I}-C_n)(x^{n+1}-x^n) \leq 0$ may be numerically inefficient, and it also isn't strictly necessary for the algorithm to converge. To simplify the description of the algorithm, suppose that $C_n = c_n C$ for some scalar $c_n > 0$ and symmetric positive definite C. Then as c_n gets larger, the method becomes more like explicit gradient projection with small time steps. This can be slow to converge as well as more prone to converging to bad local minima. However, the method still converges as long as each c_n is chosen so that the x^{n+1} update decreases F sufficiently. Therefore we want to dynamically choose $c_n \geq 0$ to be as small as possible such that the x^{n+1} update given by (3.7) decreases F by a sufficient amount, namely,

$$F(x^{n+1}) - F(x^n) \leq \sigma \left[(x^{n+1}-x^n)^T \left(\frac{1}{2}A^TA + C_n \right) (x^{n+1}-x^n) + (x^{n+1}-x^n)^T \nabla F(x^n) \right]$$

for some $\sigma \in (0,1]$. Additionally, we want to ensure that the modulus of strong convexity of the quadratic objective in (3.7) is large enough by requiring the smallest eigenvalue of $\frac{1}{2}A^TA + C_n$ to be greater than or equal to some $\rho > 0$. The following is an algorithm for solving (3.1) as well as a dynamic update scheme for $C_n = c_n C$ that is similar to Armijo line search but designed to reduce the number of times that the solution to the quadratic problem has to be rejected for not decreasing F sufficiently.

Algorithm 1. Scaled gradient projection for solving (3.1) under Assumption 1.

Define $x^0 \in X$, $c_0 > 0$, $\sigma \in (0,1]$, $\epsilon > 0$, $\rho > 0$, $\xi_1 > 1$, $\xi_2 > 1$ and set $n = 0$.

`while` $n = 0$ or $\|x^n - x^{n-1}\|_\infty > \epsilon$

$$y = \arg\min_{x \in X} (x-x^n)^T \left(\frac{1}{2}A^TA + c_n C \right)(x-x^n) + (x-x^n)^T \nabla F(x^n)$$

`if` $F(y) - F(x^n) > \sigma \left[(y-x^n)^T \left(\frac{1}{2}A^TA + c_n C \right)(y-x^n) + (y-x^n)^T \nabla F(x^n) \right]$

$$c_n = \xi_2 c_n$$

`else`

$$x^{n+1} = y$$

$$c_{n+1} = \begin{cases} \frac{c_n}{\xi_1} & \text{if smallest eigenvalue of } \frac{c_n}{\xi_1}C + \frac{1}{2}A^TA \text{ is greater than } \rho \\ c_n & \text{otherwise} \end{cases}$$

$$n = n+1$$

```
    end if

end while
```

It is not necessary to impose an upper bound on c_n in Algorithm 1 even though we want it to be bounded. The reason for this is because once $c_n \geq \lambda_R^{\max} - \frac{1}{2}\lambda_R^{\min}$, F will be sufficiently decreased for any choice of $\sigma \in (0, 1]$, so c_n is effectively bounded by $\xi_2(\lambda_R^{\max} - \frac{1}{2}\lambda_R^{\min})$.

Under Assumption 2 it is much more straightforward to derive an estimate analogous to Proposition 3.1. Concavity of $R(x)$ immediately implies

$$R(y) \leq R(x) + (y - x)^T \nabla R(x).$$

Adding to this the expression

$$\frac{1}{2}\|Ay - b\|^2 = \frac{1}{2}\|Ax - b\|^2 + (y - x)^T(A^T Ax - A^T b) + \frac{1}{2}(y - x)^T A^T A(y - x)$$

yields

$$F(y) - F(x) \leq (y - x)^T \frac{1}{2} A^T A(y - x) + (y - x)^T \nabla F(x) \tag{3.8}$$

for $x, y \in X$. Moreover, the estimate still holds if we add $(y - x)^T C(y - x)$ to the right-hand side for any positive semidefinite matrix C. We are again led to iterate (3.7) to decrease F, and in this case C_n need only be included to ensure that $A^T A + 2C_n$ is positive definite. We can let $C_n = C$ since the dependence on n is no longer necessary. We can choose any C such that the smallest eigenvalue of $C + \frac{1}{2}A^T A$ is greater than $\rho > 0$, but it is still preferable to choose C as small as is numerically practical.

Algorithm 2. Scaled gradient projection for solving (3.1) under Assumption 2.

Define $x^0 \in X$, C symmetric positive definite, and $\epsilon > 0$.

```
while  n = 0  or  ‖x^n − x^{n−1}‖_∞ > ε
```

$$x^{n+1} = \arg\min_{x \in X} (x - x^n)^T \left(\frac{1}{2}A^T A + C\right)(x - x^n) + (x - x^n)^T \nabla F(x^n) \tag{3.9}$$

$$n = n + 1$$

```
end while
```

Since the objective in (3.9) is zero at $x = x^n$, the minimum value is less than or equal to zero, and so $F(x^{n+1}) \leq F(x^n)$ by (3.8). Algorithm 2 is also equivalent to iterating

$$x^{n+1} = \arg\min_{x \in X} \frac{1}{2}\|Ax - b\|^2 + \|x\|_C^2 + x^T(\nabla R(x^n) - 2Cx^n),$$

which can be seen as an application of the simplified DC algorithm from [57] to $F(x) = (\frac{1}{2}\|Ax-b\|^2 + \|x\|_C^2) - (-R(x) + \|x\|_C^2)$. The DC method in [57] is more general and doesn't require the convex and concave functions to be differentiable.

With many connections to classical algorithms, existing convergence results can be applied to argue that limit points of the iterates $\{x^n\}$ of Algorithms 2 and 1 are stationary points of (3.1). We still choose to include a convergence analysis for clarity because our assumptions allow us to give a simple and intuitive argument. The following analysis is for Algorithm 1 under Assumption 1. However, if we replace C_n with C and σ with 1, then it applies equally well to Algorithm 2 under Assumption 2. We proceed by showing that the sequence $\{x^n\}$ is bounded, $\|x^{n+1}-x^n\| \to 0$, and limit points of $\{x^n\}$ are stationary points of (3.1) satisfying the necessary local optimality condition (2.3).

Lemma 3.3. *The sequence of iterates $\{x^n\}$ generated by Algorithm 1 is bounded.*

Proof. Since $F(x^n)$ is nonincreasing, $x^n \in \{x \in X : F(x) \le F(x^0)\}$, which is a bounded set by assumption. ∎

Lemma 3.4. *Let $\{x^n\}$ be the sequence of iterates generated by Algorithm 1. Then $\|x^{n+1}-x^n\| \to 0$.*

Proof. Since $\{F(x^n)\}$ is bounded below and nonincreasing, it converges. By construction, x^{n+1} satisfies

$$-\left[(x^{n+1}-x^n)^T\left(\frac{1}{2}A^TA + C_n\right)(x^{n+1}-x^n) + (x^{n+1}-x^n)^T\nabla F(x^n)\right] \le \frac{1}{\sigma}(F(x^n)-F(x^{n+1})).$$

By the optimality condition for (3.7),

$$(y-x^{n+1})^T\left((A^TA+2C_n)(x^{n+1}-x^n) + \nabla F(x^n)\right) \ge 0 \qquad \forall y \in X.$$

In particular, we can take $y = x^n$, which implies

$$(x^{n+1}-x^n)^T(A^TA+2C_n)(x^{n+1}-x^n) \le -(x^{n+1}-x^n)^T\nabla F(x^n).$$

Thus

$$(x^{n+1}-x^n)^T\left(\frac{1}{2}A^TA + C_n\right)(x^{n+1}-x^n) \le \frac{1}{\sigma}(F(x^n)-F(x^{n+1})).$$

Since the eigenvalues of $\frac{1}{2}A^TA + C_n$ are bounded below by $\rho > 0$, we have that

$$\rho\|x^{n+1}-x^n\|^2 \le \frac{1}{\sigma}(F(x^n)-F(x^{n+1})).$$

The result follows from noting that

$$\lim_{n\to\infty}\|x^{n+1}-x^n\|^2 \le \lim_{n\to\infty}\frac{1}{\sigma\rho}(F(x^n)-F(x^{n+1})),$$

which equals 0 since $\{F(x^n)\}$ converges. ∎

Proposition 3.5. *Any limit point x^* of the sequence of iterates $\{x^n\}$ generated by Algorithm 1 satisfies $(y - x^*)^T \nabla F(x^*) \geq 0$ for all $y \in X$, which means that x^* is a stationary point of (3.1).*

Proof. Let x^* be a limit point of $\{x^n\}$. Since $\{x^n\}$ is bounded, such a point exists. Let $\{x^{n_k}\}$ be a subsequence that converges to x^*. Since $\|x^{n+1} - x^n\| \to 0$, we also have that $x^{n_k+1} \to x^*$. Recalling the optimality condition for (3.7),

$$\begin{aligned} 0 &\leq (y - x^{n_k+1})^T \left((A^T A + 2C_{n_k})(x^{n_k+1} - x^{n_k}) + \nabla F(x^{n_k}) \right) \\ &\leq \|y - x^{n_k+1}\| \|A^T A + 2C_{n_k}\| \|x^{n_k+1} - x^{n_k}\| + (y - x^{n_k+1})^T \nabla F(x^{n_k}) \qquad \forall y \in X. \end{aligned}$$

Following [7], proceed by taking the limit along the subsequence as $n_k \to \infty$. We have that

$$\|y - x^{n_k+1}\| \|x^{n_k+1} - x^{n_k}\| \|A^T A + 2C_{n_k}\| \to 0$$

since $\|x^{n_k+1} - x^{n_k}\| \to 0$ and $\|A^T A + 2C_{n_k}\|$ is bounded. By continuity of ∇F we get that

$$(y - x^*)^T \nabla F(x^*) \geq 0 \qquad \forall y \in X. \qquad \blacksquare$$

Each iteration requires minimizing a strongly convex quadratic function over the set X as defined in (3.7). Many methods can be used to solve this, and we want to choose one that is as robust as possible to poor conditioning of $\frac{1}{2}A^T A + C_n$. For example, gradient projection works theoretically and even converges at a linear rate, but it can still be impractically slow. A better choice here is to use the alternating direction method of multipliers (ADMM) [24, 25], which alternately solves a linear system involving $\frac{1}{2}A^T A + C_n$ and projects onto the constraint set. Applied to Problem 2, this is essentially the same as the application of split Bregman [26] to solve an NNLS model for hyperspectral unmixing in [56]. We consider separately the application of ADMM to Problems 1 and 2. The application to Problem 2 is simpler.

For Problem 2, (3.7) can be written as

$$x^{n+1} = \arg\min_{x \geq 0} (x - x^n)^T \left(\frac{1}{2} A^T A + C_n \right) (x - x^n) + (x - x^n)^T \nabla F_S(x^n).$$

To apply ADMM, we can first reformulate the problem as

$$\min_{u,v} g_{\geq 0}(v) + (u - x^n)^T \left(\frac{1}{2} A^T A + C_n \right) (u - x^n) + (u - x^n)^T \nabla F_S(x^n) \quad \text{such that} \quad u = v, \tag{3.10}$$

where g is an indicator function for the constraint defined by

$$g_{\geq 0}(v) = \begin{cases} 0, & v \geq 0, \\ \infty, & \text{otherwise.} \end{cases}$$

Introduce a Lagrange multiplier p, and define a Lagrangian

$$L(u, v, p) = g_{\geq 0}(v) + (u - x^n)^T \left(\frac{1}{2} A^T A + C_n \right) (u - x^n) + (u - x^n)^T \nabla F_S(x^n) + p^T (u - v) \tag{3.11}$$

and augmented Lagrangian

$$L_\delta(u,v,p) = L(u,v,p) + \frac{\delta}{2}\|u-v\|^2,$$

where $\delta > 0$. ADMM finds a saddle point

$$L(u^*,v^*,p) \le L(u^*,v^*,p^*) \le L(u,v,p^*) \qquad \forall u,v,p$$

by alternately minimizing L_δ with respect to u, minimizing with respect to v, and updating the dual variable p. Having found a saddle point of L, (u^*,v^*) will be a solution to (3.10) and we can take v^* to be the solution to (3.7). The explicit ADMM iterations are described in Algorithm 3. Here $\Pi_{\ge 0}$ denotes the orthogonal projection onto the nonnegative orthant.

Algorithm 3. ADMM for solving convex subproblem for Problem 2.

Define $\delta > 0$, v^0, and p^0 arbitrarily, and let $k = 0$.

`while` not converged

$$u^{k+1} = x^n + (A^TA + 2C_n + \delta \mathrm{I})^{-1}\left(\delta(v^k - x^n) - p^k - \nabla F_S(x^n)\right)$$
$$v^{k+1} = \Pi_{\ge 0}\left(u^{k+1} + \frac{p^k}{\delta}\right)$$
$$p^{k+1} = p^k + \delta(u^{k+1} - v^{k+1})$$
$$k = k+1$$

`end while`

For Problem 2, (3.7) can be written as

$$(x^{n+1},d^{n+1}) = \arg\min_{x,d}(x-x^n)^T\left(\frac{1}{2}A^TA + C_n^x\right)(x-x^n) + (d-d^n)^TC_n^d(d-d^n) + (x-x^n)^T\nabla_x F_H(x^n,d^n) + (d-d^n)^T\nabla_d F_H(x^n,d^n).$$

Here, ∇_x and ∇_d represent the gradients with respect to x and d, respectively. The matrix C_n is assumed to be of the form

$$C_n = \begin{bmatrix} C_n^x & 0 \\ 0 & C_n^d \end{bmatrix},$$

with C_n^d a diagonal matrix. It is helpful to represent the constraints in terms of convex sets defined by

$$X_{\epsilon_j} = \left\{ \begin{bmatrix} x_j \\ d_j \end{bmatrix} \in \mathbb{R}^{m_j+1} : \|x_j\|_1 + d_j \ge \epsilon_j, \qquad x_j \ge 0, \qquad d_j \ge 0 \right\}, \quad j = 1,\dots,M,$$

$$X_\beta = \left\{ d \in \mathbb{R}^M : \sum_{j=1}^{M} \frac{d_j}{\beta_j} \le M - r, \qquad d_j \ge 0 \right\},$$

with indicator functions $g_{X_{\epsilon_j}}$ and g_{X_β} for these sets.

Let u and w represent x and d. Then by adding splitting variables $v_x = u$ and $v_d = w$ we can reformulate the problem as

$$\min_{u,w,v_x,v_d} \sum_j g_{X_{\epsilon_j}}(v_{xj}, v_{dj}) + g_{X_\beta}(w) + (u - x^n)^T \left(\frac{1}{2}A^T A + C_n^x\right)(u - x^n) + (w - d^n)^T C_n^d (w - d^n)$$
$$+ (x - x^n)^T \nabla_x F_H(x^n, d^n) + (w - d^n)^T \nabla_d F_H(x^n, d^n) \quad \text{such that} \quad v_x = u, \quad v_d = w.$$

Adding Lagrange multipliers p_x and p_d for the linear constraints, we can define the augmented Lagrangian

$$L_\delta(u, w, v_x, v_d, p_x, p_d) = \sum_j g_{X_{\epsilon_j}}(v_{xj}, v_{dj}) + g_{X_\beta}(w) + (u - x^n)^T \left(\frac{1}{2}A^T A + C_n^x\right)(u - x^n)$$
$$+ (w - d^n)^T C_n^d (w - d^n) + (x - x^n)^T \nabla_x F_H(x^n, d^n) + (w - d^n)^T \nabla_d F_H(x^n, d^n)$$
$$+ p_x^T (u - v_x) + p_d^T (w - v_d) + \frac{\delta}{2}\|u - v_x\|^2 + \frac{\delta}{2}\|w - v_d\|^2.$$

Each ADMM iteration alternately minimizes L_δ first with respect to (u, w) and then with respect to (v_x, v_d) before updating the dual variables (p_x, p_d). The explicit iterations are described in Algorithm 4.

Algorithm 4. ADMM for solving convex subproblem for Problem 1.

Define $\delta > 0$, v_x^0, v_d^0, p_x^0, and p_d^0 arbitrarily, and let $k = 0$.
Define the weights β in the projection Π_{X_β} by $\beta_j = (\epsilon_j \sqrt{(2C_n^d + \delta \mathrm{I})_{j,j}})^{-1}$, $j = 1, \ldots, M$.

`while` not converged

$$u^{k+1} = x^n + (A^T A + 2C_n^x + \delta \mathrm{I})^{-1} \left(\delta(v_x^k - x^n) - p_x^k - \nabla_x F_H(x^n, d^n)\right)$$
$$w^{k+1} = (2C_n^d + \delta \mathrm{I})^{-\frac{1}{2}} \Pi_{X_\beta} \left((2C_n^d + \delta \mathrm{I})^{-\frac{1}{2}} (\delta v_d^k - p_d^k - \nabla_d F_H(x^n, d^n) + 2C_n^d)\right)$$
$$\begin{bmatrix} v_{xj} \\ v_{dj} \end{bmatrix}^{k+1} = \Pi_{X_{\epsilon_j}} \left(\begin{bmatrix} u_j^{k+1} + \frac{{p_x}_j^k}{\delta} \\ w_j^{k+1} + \frac{p_{d_j}^k}{\delta} \end{bmatrix} \right), \quad j = 1, \ldots, M$$
$$p_x^{k+1} = p_x^k + \delta(u^{k+1} - v_x^{k+1})$$
$$p_d^{k+1} = p_d^k + \delta(w^{k+1} - v_d^{k+1})$$
$$k = k + 1$$

`end while`

We stop iterating and let $x^{n+1} = v_x$ and $d^{n+1} = v_d$ once the relative errors of the primal and dual variables are sufficiently small. The projections Π_{X_β} and $\Pi_{X_{\epsilon_j}}$ can be efficiently computed by combining projections onto the nonnegative orthant and projections onto the appropriate simplices. These can in principle be computed in linear time [10], although we use a method that is simpler to implement and is still only $O(n \log n)$ in the dimension of the vector being projected.

Since (3.7) is a standard quadratic program, a huge variety of other methods besides ADMM could also be applied. Variants of Newton's method on a bound-constrained Karush–Kuhn–Tucker system might work well if we find that we need to solve the convex subproblems to very high accuracy. For the above applications of ADMM to be practical, the linear system involving $(A^TA + 2C_n + \delta \mathrm{I})$ should not be too difficult to solve, and δ should be well chosen. It may sometimes be helpful to use the Woodbury formula, $c(A^TA + c\mathrm{I})^{-1} = \mathrm{I} - A^T(c\mathrm{I} + AA^T)^{-1}A$, which means that we can choose to work with A^TA or AA^T, whichever is smaller. Additionally, the linear systems could be approximately solved by iterative methods such as preconditioned conjugate gradient. It may also be worthwhile to consider primal dual methods that only require matrix multiplications [14, 19]. The simplest alternative might be to apply gradient projection directly to (3.1). This can be thought of as applying the DC method to a different convex-concave splitting of F, namely, $F(x) = (\|x\|_C^2) - (\|x\|_C^2 - \frac{1}{2}\|Ax - b\|^2 - R(x))$ for sufficiently large C [58], but gradient projection may be too inefficient when A is ill-conditioned.

4. Applications.

In this section we introduce four specific applications related to DOAS analysis and hyperspectral unmixing. We show how to model these problems in the form of (3.1) so that the algorithms from section 3 can be applied.

4.1. DOAS analysis.

The goal of DOAS is to estimate the concentrations of gases in a mixture by measuring over a range of wavelengths the reduction in the intensity of light shined through it. A thorough summary of the procedure and analysis can be found in [50].

Beer's law can be used to estimate the attenuation of light intensity due to absorption. Assuming that the average gas concentration c is not too large, Beer's law relates the transmitted intensity $I(\lambda)$ to the initial intensity $I_0(\lambda)$ by

$$I(\lambda) = I_0(\lambda) \exp^{-\sigma(\lambda)cL}, \tag{4.1}$$

where λ is wavelength, $\sigma(\lambda)$ is the characteristic absorption spectra for the absorbing gas, and L is the light path length.

If the density of the absorbing gas is not constant, we should instead integrate over the light path, replacing $\exp^{-\sigma(\lambda)cL}$ by $\exp^{-\sigma(\lambda)\int_0^L c(l)dl}$. For simplicity, we will assume that the concentration is approximately constant. We will also denote the product of concentration and path length, cL, by a.

When multiple absorbing gases are present, $a\sigma(\lambda)$ can be replaced by a linear combination of the characteristic absorption spectra of the gases, and Beer's law can be written as

$$I(\lambda) = I_0(\lambda) \exp^{-\sum_j a_j \sigma_j(\lambda)}.$$

Additionally taking into account the reduction of light intensity due to scattering, com-

bined into a single term $\epsilon(\lambda)$, Beer's law becomes

$$I(\lambda) = I_0(\lambda) \exp^{-\sum_j a_j \sigma_j(\lambda) - \epsilon(\lambda)}.$$

The key idea behind DOAS is that it is not necessary to explicitly model effects such as scattering, as long as they vary smoothly enough with wavelength to be removed by high pass filtering that, loosely speaking, removes the broad structures and keeps the narrow structures. We will assume that $\epsilon(\lambda)$ is smooth. Additionally, we can assume that $I_0(\lambda)$, if not known, is also smooth. The absorption spectra $\sigma_j(\lambda)$ can be considered to be a sum of a broad part (smooth) and a narrow part, $\sigma_j = \sigma_j^{\text{broad}} + \sigma_j^{\text{narrow}}$. Since σ_j^{narrow} represents the only narrow structure in the entire model, the main idea is to isolate it by taking the log of the intensity and applying high pass filtering or any other procedure, such as polynomial fitting, that subtracts a smooth background from the data. The given reference spectra should already have had their broad parts subtracted, but it may not have been done consistently, so we will combine σ_j^{broad} and $\epsilon(\lambda)$ into a single term $B(\lambda)$. We will also denote the given reference spectra by y_j, which again are already assumed to be approximately high pass filtered versions of the true absorption spectra σ_j. With these notational changes, Beer's law becomes

$$I(\lambda) = I_0(\lambda) \exp^{-\sum_j a_j y_j(\lambda) - B(\lambda)}. \tag{4.2}$$

In practice, measurement errors must also be modeled. We therefore consider multiplying the right-hand side of (4.2) by $s(\lambda)$, representing wavelength-dependent sensitivity. Assuming that $s(\lambda) \approx 1$ and varies smoothly with λ, we can absorb it into $B(\lambda)$. Measurements may also be corrupted by convolution with an instrument function $h(\lambda)$, but for simplicity we will assume that this effect is negligible and not include convolution with h in the model. Let $J(\lambda) = -\ln(I(\lambda))$. This is what we will consider to be the given data. By taking the log, the previous model simplifies to

$$J(\lambda) = -\ln(I_0(\lambda)) + \sum_j a_j y_j(\lambda) + B(\lambda) + \eta(\lambda),$$

where $\eta(\lambda)$ represents the log of multiplicative noise, which we will model as being approximately white Gaussian noise.

Since $I_0(\lambda)$ is assumed to be smooth, it can also be absorbed into the $B(\lambda)$ component, yielding the data model

$$J(\lambda) = \sum_j a_j y_j(\lambda) + B(\lambda) + \eta(\lambda). \tag{4.3}$$

4.1.1. DOAS analysis with wavelength misalignment. A challenging complication in practice is wavelength misalignment; i.e., the nominal wavelengths in the measurement $J(\lambda)$ may not correspond exactly to those in the basis $y_j(\lambda)$. We must allow for small, often approximately linear deformations $v_j(\lambda)$ so that $y_j(\lambda + v_j(\lambda))$ are all aligned with the data $J(\lambda)$. Taking into account wavelength misalignment, the data model becomes

$$J(\lambda) = \sum_j a_j y_j(\lambda + v_j(\lambda)) + B(\lambda) + \eta(\lambda). \tag{4.4}$$

To first focus on the alignment aspect of this problem, assume that $B(\lambda)$ is negligible, having somehow been consistently removed from the data and references by high pass filtering or polynomial subtraction. Then, given the data $J(\lambda)$ and reference spectra $\{y_j(\lambda)\}$, we want to estimate the fitting coefficients $\{a_j\}$ and the deformations $\{v_j(\lambda)\}$ from the linear model,

$$J(\lambda) = \sum_{j=1}^{M} a_j y_j\big(\lambda + v_j(\lambda)\big) + \eta(\lambda), \tag{4.5}$$

where M is the total number of gases to be considered.

Inspired by the idea of using a set of modified bases for image deconvolution [44], we construct a dictionary by deforming each y_j with a set of possible deformations. Specifically, since the deformations can be well approximated by linear functions, i.e., $v_j(\lambda) = p_j\lambda + q_j$, we enumerate all the possible deformations by choosing p_j, q_j from two predetermined sets $\{P_1, \dots, P_K\}$, $\{Q_1, \dots, Q_L\}$. Let A_j be a matrix whose columns are deformations of the jth reference $y_j(\lambda)$, i.e., $y_j(\lambda + P_k\lambda + Q_l)$ for $k = 1, \dots, K$ and $l = 1, \dots, L$. Then we can rewrite the model (4.5) in terms of a matrix-vector form,

$$J = [A_1, \dots, A_M] \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} + \eta, \tag{4.6}$$

where $x_j \in \mathbb{R}^{KL}$ and $J \in \mathbb{R}^W$.

We propose the following minimization model:

$$\begin{aligned} &\arg\min_{x_j} \tfrac{1}{2} \left\| J - [A_1, \dots, A_M] \begin{bmatrix} x_1 \\ \vdots \\ x_M \end{bmatrix} \right\|^2 \\ &\text{such that} \quad x_j \geqslant 0,\ \|x_j\|_0 \leqslant 1, \qquad j = 1, \dots, M. \end{aligned} \tag{4.7}$$

The second constraint in (4.7) is to force each x_j to have at most one nonzero element. Having $\|x_j\|_0 = 1$ indicates the existence of the gas with a spectrum y_j. Its nonzero index corresponds to the selected deformation, and its magnitude corresponds to the concentration of the gas. This l_0 constraint makes the problem NP-hard. A direct approach is the penalty decomposition method proposed in [45], which we will compare to in section 5. Our approach is to replace the l_0 constraint on each group with intrasparsity penalties defined by H_j in (2.4) or S_j^ϵ in (2.8), putting the problem in the form of Problem 1 or 2. The intrasparsity parameters γ_j should be chosen large enough to enforce 1-sparsity within groups, and in the absence of any intergroup sparsity assumptions we can set $\gamma_0 = 0$.

4.1.2. DOAS with background model. To incorporate the background term from (4.4), we will add $B \in \mathbb{R}^W$ as an additional unknown and also add a quadratic penalty $\frac{\alpha}{2}\|QB\|^2$ to penalize a lack of smoothness of B. This leads to the model

$$\min_{x \in X, B} \frac{1}{2}\|Ax + B - J\|^2 + \frac{\alpha}{2}\|QB\|^2 + R(x),$$

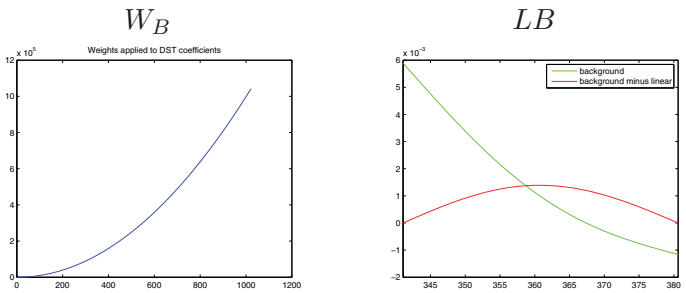

Figure 4. *Functions used to define background penalty.*

where R includes our choice of intrasparsity penalties on x. This can be rewritten as

$$\min_{x\in X,B} \frac{1}{2}\left\|\begin{bmatrix} A & \mathrm{I} \\ 0 & \sqrt{\alpha}Q \end{bmatrix}\begin{bmatrix} x \\ B \end{bmatrix} - \begin{bmatrix} J \\ 0 \end{bmatrix}\right\|^2 + R(x). \tag{4.8}$$

This model has the general form of (3.1) with the two-by-two block matrix interpreted as A and $\begin{bmatrix} J \\ 0 \end{bmatrix}$ interpreted as b. Moreover, we can concatenate B and the M groups x_j by considering B to be group x_{M+1} and setting $\gamma_{M+1} = 0$ so that no sparsity penalty acts on the background component. In this way, we see that the algorithms presented in section 3 can be directly applied to (4.8).

It remains to define the matrix Q used in the penalty to enforce smoothness of the estimated background. A possible strategy is to work with the discrete Fourier transform or discrete cosine transform of B and penalize high frequency coefficients. Although B should be smooth, it is unlikely to satisfy Neumann or periodic boundary conditions, so based on an idea in [52], we will work with B minus the linear function that interpolates its endpoints. Let $L \in \mathbb{R}^{W\times W}$ be the matrix representation of the linear operator that takes the difference of B and its linear interpolant. Since LB satisfies zero boundary conditions and its odd periodic extension should be smooth, its discrete sine transform (DST) coefficients should rapidly decay. So we can penalize the high frequency DST coefficients of LB to encourage smoothness of B. Let Γ denote the DST, and let W_B be a diagonal matrix of positive weights that are larger for higher frequencies. An effective choice is $\mathrm{diag}(W_B)_i = i^2$, since the index $i = 0, \ldots, W-1$ is proportional to frequency. We then define $Q = W_B\Gamma L$ in (4.8) and can adjust the strength of this smoothing penalty by changing the single parameter $\alpha > 0$. Figure 4 shows the weights W_B and the result LB of subtracting from B the line interpolating its endpoints.

4.2. Hyperspectral image analysis. Hyperspectral images record high resolution spectral information at each pixel of an image. This large amount of spectral data makes it possible to identify materials based on their spectral signatures. A hyperspectral image can be represented as a matrix $Y \in \mathbb{R}^{W\times P}$, where P is the number of pixels and W is the number of spectral bands.

Due to low spatial resolution or finely mixed materials, each pixel can contain multiple different materials. The spectral data measured at each pixel, according to a linear mixing model, is assumed to be a nonnegative linear combination of spectral signatures of pure

materials, which are called endmembers. The list of known endmembers can be represented as the columns of a matrix $A \in \mathbb{R}^{W\times N}$.

The goal of hyperspectral unmixing is to determine the abundances of different materials at each pixel. Given Y, and if A is also known, the goal is then to determine an abundance matrix $S \in \mathbb{R}^{N\times P}$ with $S_{i,j} \geq 0$. Each row of S is interpretable as an image that shows the abundance of one particular material at every pixel. Mixtures are often assumed to involve only very few of the possible materials, so the columns of S are often additionally assumed to be sparse.

4.2.1. Sparse hyperspectral unmixing. A simple but effective approach for hyperspectral unmixing is NNLS, which solves

$$\min_{S\geq 0} \|Y - AS\|_F^2,$$

where F denotes the Frobenius norm. Many other tools have also been used to encourage additional sparsity of S, such as l_1 minimization and variants of matching pursuit [29, 56, 35, 27]. If no spatial correlations are assumed, the unmixing problem can be solved at each pixel independently. We can also add one of the nonconvex intersparsity penalties defined by H_0 in (2.4) or S_0^ϵ in (2.8). The resulting problem can be written in the form

$$\min_{x_p\geq 0} \frac{1}{2}\|Ax_p - b_p\|^2 + R(x_p), \tag{4.9}$$

where x_p is the pth column of S and b_p is the pth column of Y. We can define $R(x_p)$ to equal $H_0(x_p)$ or $S_0^\epsilon(x_p)$, putting (4.9) in the general form of (3.1).

4.2.2. Structured sparse hyperspectral unmixing. In hyperspectral unmixing applications, the dictionary of endmembers is usually not known precisely. There are many methods for learning endmembers from a hyperspectral image such as N-FINDR [62], vertex component analysis (VCA) [48], NMF [49], Bayesian methods [66, 13], and convex optimization [18]. However, here we are interested in the case where we have a large library of measured reference endmembers including multiple references for each expected material measured under different conditions. The resulting dictionary A is assumed to have the group structure $[A_1, \ldots, A_M]$, where each group A_j contains different references for the same jth material.

There are several reasons that we don't want to use the sparse unmixing methods of section 4.2.1 when A contains a large library of references defined in this way. Such a matrix A with many nearly redundant references will likely have high coherence. This creates a challenge for existing methods. The grouped structure of A also means that we want to enforce a structured sparsity assumption on the columns of S. The linear combination of endmembers at any particular pixel is assumed to involve at most one endmember from each group A_j. Linearly combining multiple references within a group may not be physically meaningful, since they all represent the same material. Restricting our attention to a single pixel p, we can write the pth abundance column x_p of S as

$$\begin{bmatrix} x_{1,p} \\ \vdots \\ x_{M,p} \end{bmatrix}.$$

The sparsity assumption requires each group of abundance coefficients $x_{j,p}$ to be at most 1-sparse. We can enforce this by adding sufficiently large intrasparsity penalties to the objective in (4.9) defined by $H_j(x_{j,p})$ (2.4) or $S_j^\epsilon(x_{j,p})$ (2.8).

We think it may be important to use an expanded dictionary to allow different endmembers within groups to be selected at different pixels, thus incorporating endmember variability into the unmixing process. Existing methods accomplish this in different ways, such as the piecewise convex endmember detection method in [67], which represents the spectral data as convex combinations of endmember distributions. It is observed in [67] that real hyperspectral data can be better represented using several sets of endmembers. Additionally, their better performance compared to VCA, which assumes pixel purity, on a dataset which should satisfy the pixel purity assumption, further justifies the benefit of incorporating endmember variability when unmixing.

If the same set of endmembers were valid at all pixels, we could attempt to enforce row sparsity of S using, for example, the $l_{1,\infty}$ penalty used in [18], which would encourage the data at all pixels to be representable as nonnegative linear combinations of the same small subset of endmembers. Under some circumstances, this is a reasonable assumption and could be a good approach. However, due to varying conditions, a particular reference for some material may be good at some pixels but not at others. Although atmospheric conditions are of course unlikely to change from pixel to pixel, there could be nonlinear mixing effects that make the same material appear to have different spectral signatures in different locations [39]. For instance, a nonuniform layer of dust will change the appearance of materials in different places. If this mixing with dust is nonlinear, then the resulting hyperspectral data cannot necessarily be well represented by the linear mixture model with a dust endmember added to the dictionary. In this case, by considering an expanded dictionary containing reference measurements for the materials covered by different amounts of dust, we are attempting to take into account these nonlinear mixing effects without explicitly modeling them. At different pixels, different references for the same materials can now be used when trying to best represent the data. We should point out that our approach is effective when only a small number of nonlinear effects need to be taken into account. The more spectral variability we include for each endmember, the larger the matrix A becomes. Our method is applicable when there are relatively few realizations of endmember variability in the data and these realizations are well represented in the expanded dictionary.

The overall model should contain both intra- and intersparsity penalties. In addition to the 1-sparsity assumption within groups, it is still assumed that many fewer than M materials are present at any particular pixel. The full model can again be written as (4.9) except with the addition of intrasparsity penalties. The overall sparsity penalties can be written as either

$$R(x_p, d_p) = \sum_{j=1}^{M} \gamma_j H_j(x_{j,p}, d_{j,p}) + \gamma_0 H_0(x_p)$$

or

$$R(x_p) = \sum_{j=1}^{M} \gamma_j S_j^{\epsilon_j}(x_{j,p}) + \gamma_0 S_0^{\epsilon_0}(x_p).$$

5. Numerical experiments. In this section, we evaluate the effectiveness of our implementations of Problems 1 and 2 on the four applications discussed in section 4. The simplest DOAS example with wavelength misalignment from section 4.1.1 is used to see how well the intrasparsity assumption is satisfied compared to other methods. Two convex methods that we compare to are NNLS (1.1) and a nonnegative constrained l_1 basis pursuit model like the template matching via l_1 minimization in [28]. The l_1 minimization model we use here is

$$\min_{x\geq 0} \|x\|_1 \quad \text{such that} \quad \|Ax-b\| \leq \tau. \tag{5.1}$$

We use the MATLAB function `lsqnonneg`, which is parameter free, to solve the NNLS model. We use Bregman iteration [63] to solve the l_1 minimization model. We also compare to direct l_0 minimization via penalty decomposition (Algorithm 5).

The penalty decomposition method [45] amounts to solving (4.7) by a series of minimization problems with an increasing sequence $\{\rho_k\}$. Let $x=[\mathbf{x}_1,\dots,\mathbf{x}_M]$, $y=[\mathbf{y}_1,\dots,\mathbf{y}_M]$, and iterate

$$\begin{aligned} &(x^{k+1},y^{k+1}) = \arg\min \tfrac{1}{2}\|Ax-b\|^2+\tfrac{\rho_k}{2}\|x-y\|^2 \quad \text{such that} \quad \mathbf{y}_j \geqslant 0,\ \|\mathbf{y}_j\|_0 \leqslant 1,\\ &\rho^{k+1}=\sigma\rho^k \quad (\text{for } \sigma>1). \end{aligned} \tag{5.2}$$

The pseudocode of this method is given in Algorithm 5.

Algorithm 5. A penalty decomposition method for solving (4.7).

Define $\rho>0$, $\sigma>1$, ϵ_o, ϵ_i and initialize y.

```
while ||x − y||_∞ > ε_o
    i = 1;
    while max{||x^i − x^{i−1}||_∞, ||y^i − y^{i−1}||_∞} > ε_i
        x^i = (A^T A + ρId)^{−1}(A^T b + ρy^i),
        y^i = 0
        for j = 1, ..., M
            Find the index of maximal x_j, i.e., l_j = arg max_l x_j(l).
            Set y_j(l_j) = max(x_j(l_j), 0).
        end for
        i = i + 1;
    end while
    x = x^i, y = y^i, ρ = σρ
end while
```

Algorithm 5 may require a good initialization of y or a slowly increasing ρ. If the maximum magnitude locations within each group are initially incorrect, it can get stuck at a local minimum. We consider both least squares (LS) and NNLS initializations in numerical experiments. Algorithms 1 and 2 also benefit from a good initialization for the same reason. We use

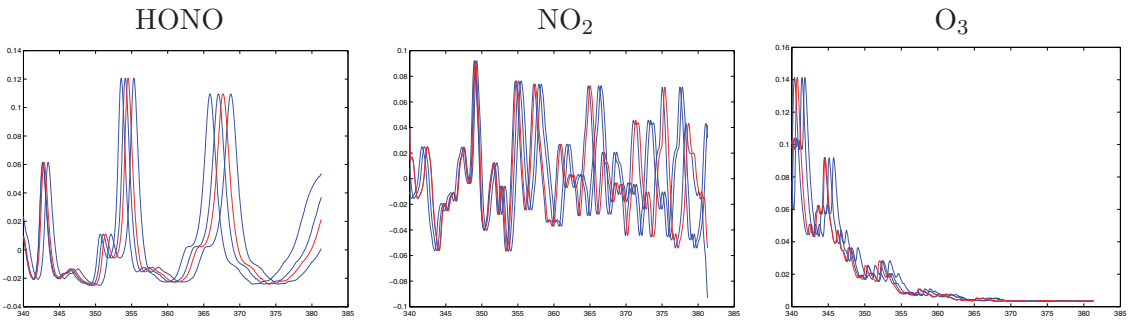

Figure 5. *For each gas, the reference spectrum is plotted in red, while three deformed spectra are in blue.*

a constant initialization, for which the first iteration of those methods is already quite similar to that of NNLS.

We also test the effectiveness of Problems 1 and 2 on the other three applications discussed in section 4. For DOAS with the included background model, we compare again to Algorithm 5. We use the sparse hyperspectral unmixing example to demonstrate the sparsifying effect of the intersparsity penalties acting without any intrasparsity penalties. We compare to the l_1 regularized unmixing model in [29] using the implementation in [56]. To illustrate the effect of the intra- and intersparsity penalties acting together, we also apply Problems 1 and 2 to a synthetic example of structured sparse hyperspectral unmixing. We compare the recovery of the ground truth abundance with and without the intrasparsity penalties.

5.1. DOAS with wavelength alignment. We generate the dictionary by taking three given reference spectra $y_j(\lambda)$ for the gases nitrous acid (HONO), nitrogen dioxide (NO_2), and ozone (O_3) and deforming each by a set of linear functions. The resulting dictionary contains $y_j(\lambda + P_k\lambda + Q_l)$ for $P_k = -1.01 + 0.01k$ $(k = 1, \ldots, 21)$, $Q_l = -1.1 + 0.1l$ $(l = 1, \ldots, 21)$, and $j = 1, 2, 3$. Each $y_j \in \mathbb{R}^W$ with $W = 1024$. The represented wavelengths in nanometers are $\lambda = 340 + 0.04038w$, $w = 0, \ldots, 1023$. We use odd reflections to extrapolate shifted references at the boundary. The choice of boundary condition should have only a small effect if the wavelength displacements are small. However, if the displacements are large, it may be a good idea to modify the data fidelity term to select only the middle wavelengths to prevent boundary artifacts from influencing the results.

There are a total of 441 linearly deformed references for each of the three groups. In Figure 5, we plot the reference spectra of HONO, NO_2, and O_3 together with several deformed examples.

In our experiments, we randomly select one element for each group with random magnitude plus additive zero mean Gaussian noise to synthesize the data term $J(\lambda) \in \mathbb{R}^W$ for $W = 1024$. Mimicking the relative magnitudes of a real DOAS dataset [22] after normalization of the dictionary, the random magnitudes are chosen to be at different orders with mean values of 1, 0.1, and 1.5 for HONO, NO_2, and O_3, respectively. We perform three experiments for which the standard deviations of the noise are 0, .005, and .05, respectively. This synthetic data is shown in Figure 6.

The parameters used in the numerical experiments are as follows. NNLS is parameter

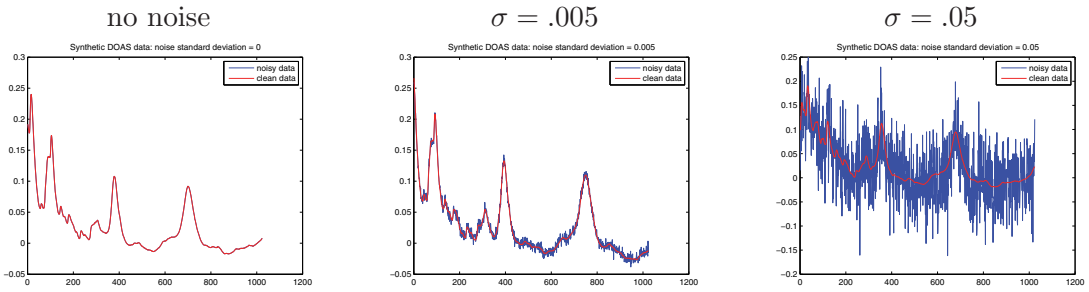

Figure 6. *Synthetic DOAS data.*

free. For the l_1 minimization method in (5.1), $\frac{\tau}{\sqrt{W}} = .001$, .005, and .05 for the experiments with noise standard deviations of 0, .005, and .05, respectively. For the direct l_0 method (Algorithm 5), the penalty parameter ρ is initially equal to .05 and increases by a factor of $\sigma = 1.2$ every iteration. The inner and outer tolerances are set at 10^{-4} and 10^{-5}, respectively. The initialization is chosen to be either an LS solution or the result of NNLS. For Problems 1 and 2 we define $\epsilon_j = .05$ for all three groups. In general this could be chosen roughly on the order of the smallest nonzero coefficient expected in the jth group. Recall that these ϵ_j are used both in the definitions of the regularized $l_1 - l_2$ penalties S_j^ϵ in Problem 2 and in the definitions of the dummy variable constraints in Problem 1. We set $\gamma_j = .1$ and $\gamma_j = .05$ for Problems 1 and 2, respectively, and for $j = 1, 2, 3$. Since there is no intersparsity penalty, $\gamma_0 = 0$. For both Algorithms 1 and 2 we set $C = 10^{-9}\mathrm{I}$. For Algorithm 1, which dynamically updates C, we set several additional parameters $\sigma = .1$, $\xi_1 = 2$, and $\xi_2 = 10$. These choices are not crucial and have more to do with the rate of convergence than the quality of the result. For both algorithms, the outer iterations are stopped when the difference in energy is less than 10^{-8}, and the inner ADMM iterations are stopped when the relative errors of the primal and dual variables are both less than 10^{-4}.

We plot results of the different methods in blue along with the ground truth solution in red. The experiments are shown in Figures 7–9.

5.2. DOAS with wavelength alignment and background estimation. We solve the model (4.8) using l_1/l_2 and regularized $l_1 - l_2$ intrasparsity penalties. These are special cases of Problems 1 and 2, respectively. Depending on which, the convex set X is either the nonnegative orthant or a subset of it. We compare the performance to the direct l_0 method (Algorithm 5) and LS. The dictionary consists of the same set of linearly deformed reference spectra for HONO, NO_2, and O_3 as in section 5.1. The data J is synthetically generated by

$$J(\lambda) = .0121y_1(\lambda) + .0011y_2(\lambda) + .0159y_3(\lambda) + \frac{2}{(\lambda - 334)^4} + \eta(\lambda),$$

where the references y_j are drawn from columns 180, 682, and 1103 of the dictionary and the last two terms represent a smooth background component and zero mean Gaussian noise having standard deviation 5.5810^{-5}. The parameter α in (4.8) is set at 10^{-5} for all the experiments.

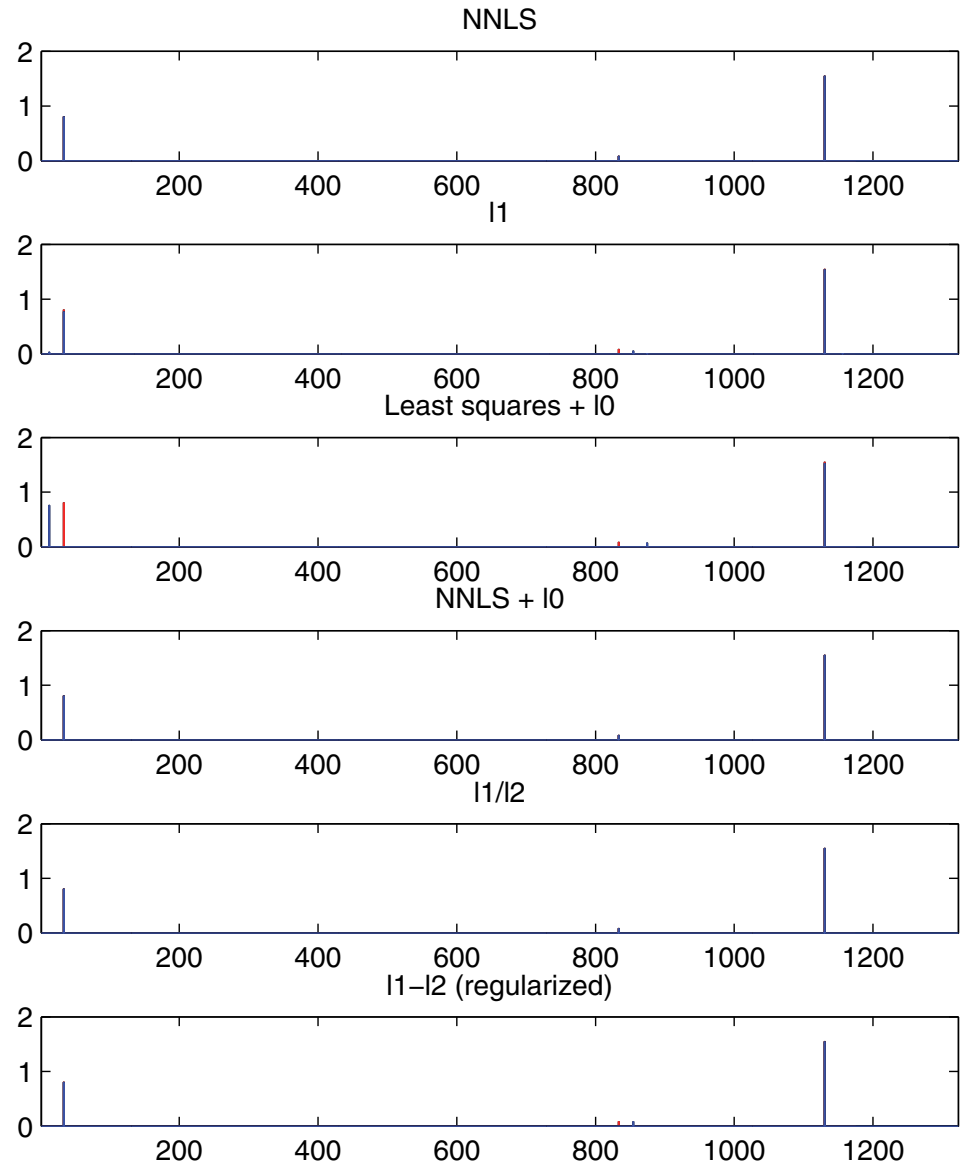

Figure 7. *Method comparisons on synthetic DOAS data without noise. Computed coefficients (blue) are plotted on top of the ground truth (red).*

The LS method for (4.8) directly solves

$$\min_{x,B} \frac{1}{2} \left\| \begin{bmatrix} A_3 & \mathrm{I} \\ 0 & \sqrt{\alpha} Q \end{bmatrix} \begin{bmatrix} x \\ B \end{bmatrix} - \begin{bmatrix} J \\ 0 \end{bmatrix} \right\|^2,$$

where A_3 has only three columns randomly chosen from the expanded dictionary A, with one chosen from each group. Results are averaged over 1000 random selections.

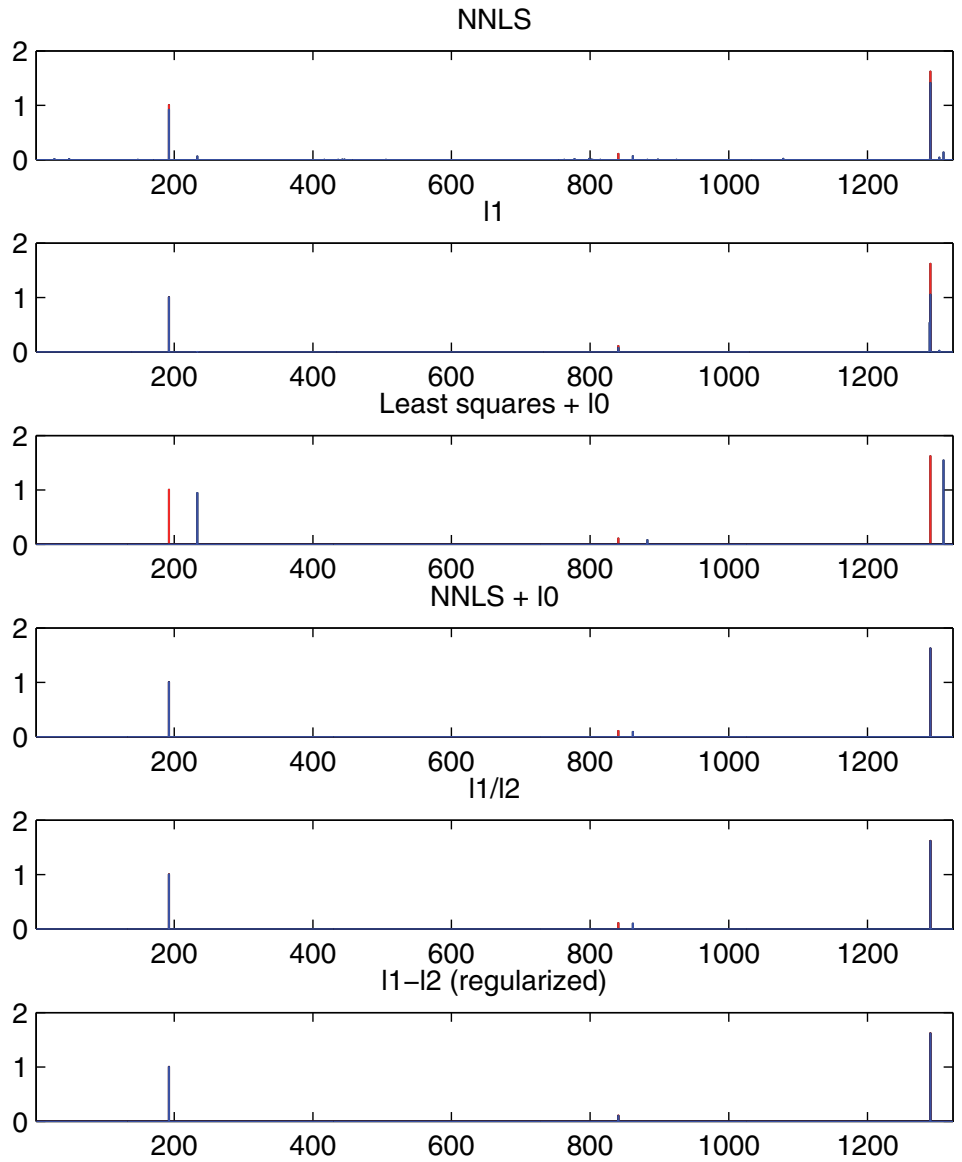

Figure 8. *Method comparisons on synthetic DOAS data:* $\sigma = .005$. *Computed coefficients (blue) are plotted on top of the ground truth (red).*

In Algorithm 5, the penalty parameter ρ starts at 10^{-6} and increases by a factor of $\sigma = 1.1$ every iteration. The inner and outer tolerances are set at 10^{-4} and 10^{-6}, respectively. The coefficients are initialized to zero.

In Algorithms 1 and 2, we treat the background as a fourth group of coefficients, after the three for each set of reference spectra. For all groups ϵ_j is set to .001. We set $\gamma_j = .001$ for $j = 1, 2, 3$, and $\gamma_4 = 0$, so no sparsity penalty is acting on the background component. We set

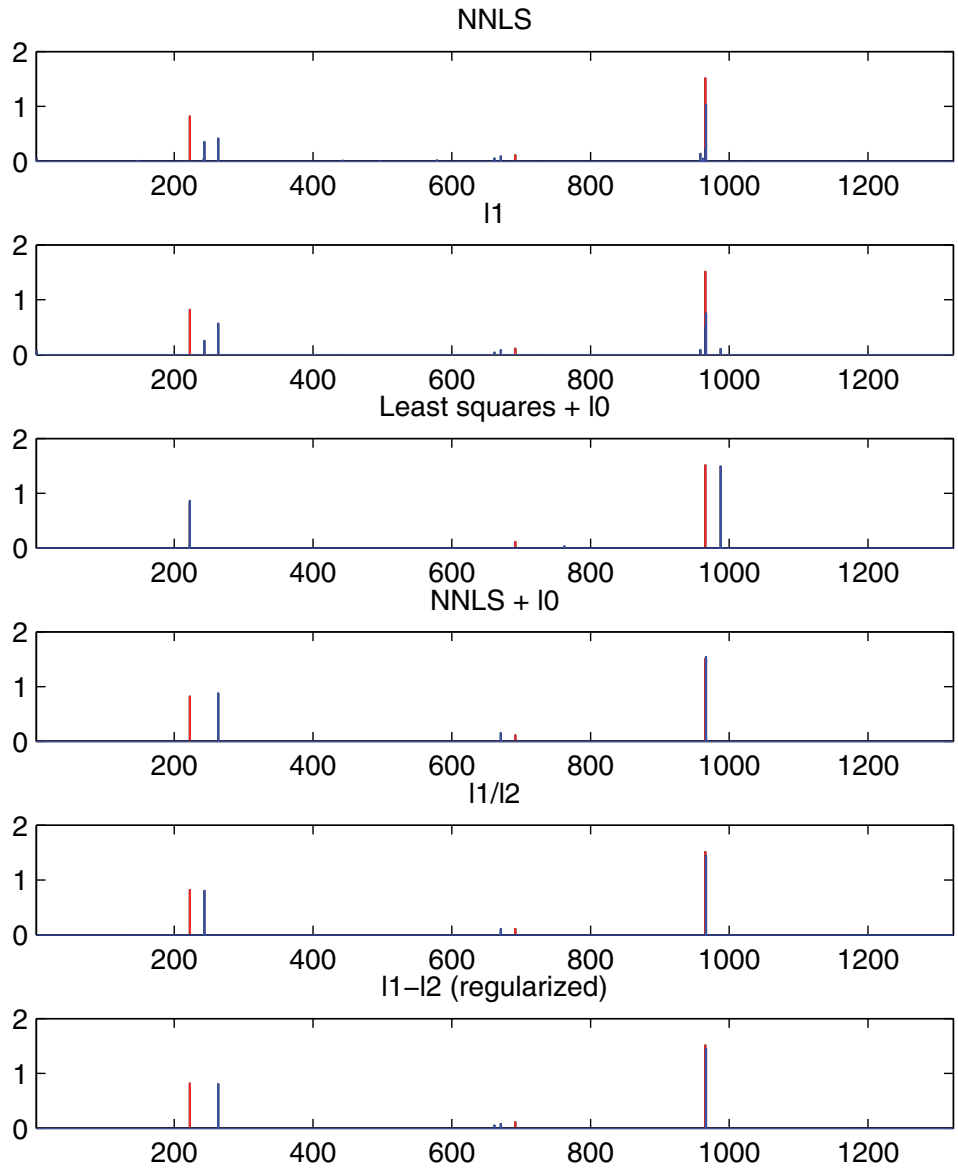

Figure 9. *Method comparisons on synthetic DOAS data: $\sigma = .05$. Computed coefficients (blue) are plotted on top of the ground truth (red).*

$C = 10^{-7}\mathrm{I}$ for Algorithm 2 and $C = 10^{-4}\mathrm{I}$ for Algorithm 1, where again we use $\sigma = .1$, $\xi_1 = 2$, and $\xi_2 = 10$. We use a constant but nonzero initialization for the coefficients x. The inner and outer iteration tolerances are the same as in section 5.1 with the inner decreased to 10^{-5}.

Figure 10 compares how closely the results of the four methods fit the data. Plotted are the synthetic data, the estimated background, each of the selected three linearly deformed

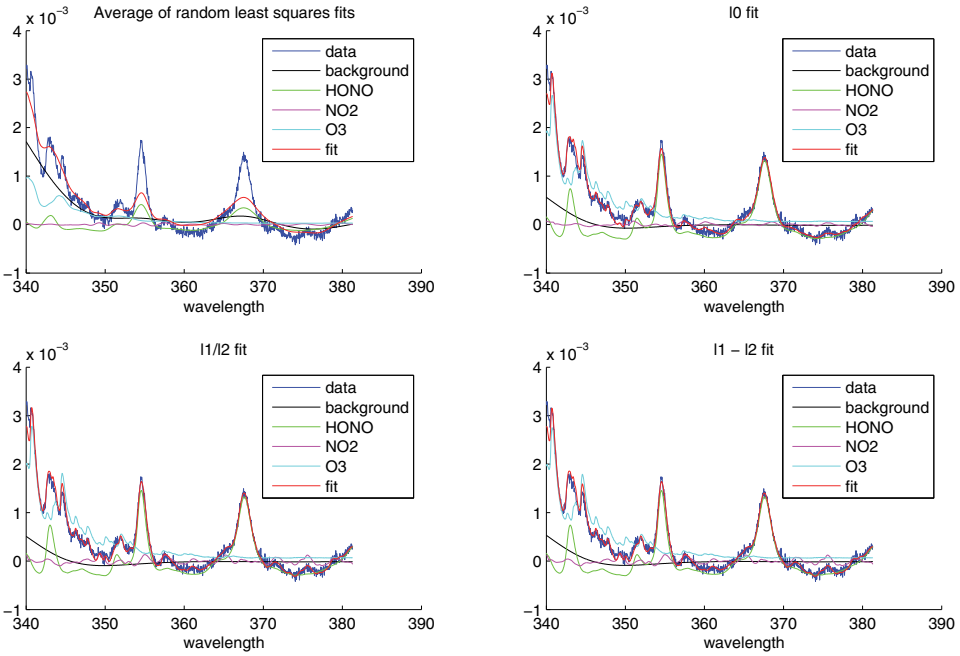

Figure 10. *Comparisons of how well the results of least squares, direct l_0, l_1/l_2, and regularized $l_1 - l_2$ fit the data.*

Table 1

Comparison of estimated fitting coefficients and displacements for DOAS with background estimation.

	Ground truth	LS	l_0	l_1/l_2	$l_1 - l_2$
a_1 (HONO coefficient)	0.01206	0.00566	0.01197	0.01203	0.01202
a_2 (NO_2 coefficient)	0.00112	0.00020	0.00081	0.00173	0.00173
a_3 (O_3 coefficient)	0.01589	0.00812	0.01884	0.01967	0.01947
v_1 (HONO displacement)	$0.01\lambda - 0.2$	N/A	$0.01\lambda - 0.2$	$0.01\lambda - 0.2$	$0.01\lambda - 0.2$
v_2 (NO_2 displacement)	$-0.01\lambda + 0.1$	N/A	$-0.09\lambda - 0.9$	$0\lambda - 0.2$	$0\lambda - 0.2$
v_3 (O_3 displacement)	$0\lambda + 0$	N/A	$0\lambda + 0$	$0\lambda + 0$	$0\lambda + 0$

reference spectra multiplied by their estimated fitting coefficients, and, finally, the sum of the references and background.

The computed coefficient magnitudes and displacements are compared to the ground truth in Table 1.

The dictionary perhaps included some unrealistically large deformations of the references. Nonetheless, the LS result shows that the coefficient magnitudes are underestimated when the alignment is incorrect. The methods for the l_0, l_1/l_2 and regularized $l_1 - l_2$ models all produced good and nearly equivalent results. All estimated the correct displacements of HONO and O_3, but not NO_2. The estimated amounts of HONO and NO_2 were correct. The amount of O_3 was overestimated by all methods. This is because there was a large background component in the O_3 reference. Even with background estimation included in the model, working with references that have been high pass filtered ahead of time should still improve accuracy.

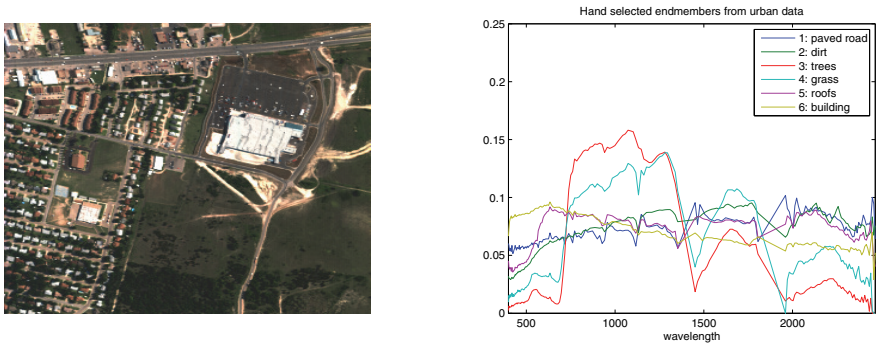

Figure 11. *Color visualization of urban hyperspectral image and hand selected endmembers.*

Although the methods for the l_0, l_1/l_2, and regularized $l_1 - l_2$ models all yielded similar solutions, they have different pros and cons regarding parameter selection and runtime. It is important that ρ not increase too quickly in the direct l_0 method. Otherwise it can get stuck at a poor solution. For this DOAS example, the resulting method required about 200 iterations and a little over 10 minutes to converge. Algorithm 1 for the l_1/l_2 model can sometimes waste effort finding splitting coefficients that yield a sufficient decrease in energy. Here it required 20 outer iterations and ran in a few minutes. Algorithm 2 required 8 outer iterations and took about a minute. Choosing γ_j too large can also cause the l_1/l_2 and $l_1 - l_2$ methods to get stuck at bad local minima. On the other hand, choosing γ_j too small may result in the group 1-sparsity condition not being satisfied, whereas it is satisfied by construction in the direct l_0 approach. Empirically, gradually increasing γ_j works well, but we have simply used fixed parameters for all of our experiments.

5.3. Hyperspectral unmixing with intersparsity penalty. We use the urban hyperspectral dataset from [2]. Each column of the data matrix $Y \in \mathbb{R}^{187\times 94249}$ represents the spectral signature measured at a pixel in the 307-by-307 urban image shown in Figure 11.

The data was processed to remove some wavelengths for which the data was corrupted, resulting in a spectral resolution reduced from 210 to 187. The six endmembers forming the columns of the dictionary A were selected by hand from pixels that appeared to be pure materials. These are also shown in Figure 11. The columns of both A and Y were normalized to have unit l_2 norm.

It is common in hyperspectral unmixing to enforce a sum to one constraint on each column of the abundance matrix S, whose entries can then be directly interpreted as proportions of materials present at each pixel. For our experiments we don't enforce this constraint, nor do we expect it to be satisfied having assumed that the data is l_2 normalized. With l_2 normalized data and endmembers, we are unmixing based on the shapes of the spectral signatures, not their magnitudes. Another reason for this assumption is that we want to compare to l_1 unmixing, which is not meaningful under a sum to one constraint but can promote sparsity otherwise. In practice, normalizing the data is like using a weighted Frobenius norm for the data fidelity penalty and may introduce bias if it's not consistent with the error model, but our focus here is on the sparsity penalties and not on the error model.

Table 2

Fraction of nonzero abundances and sum of squares error for four unmixing models.

	NNLS	l_1	l_1/l_2	$l_1 - l_2$
Fraction nonzero	0.4752	0.2683	0.2645	0.2677
Sum of squares error	1111.2	19107	1395.3	1335.6

Algorithms 1 and 2 were used to solve (4.9) with l_1/l_2 and regularized $l_1 - l_2$ intersparsity penalties, respectively. These algorithms were compared to NNLS and l_1 minimization [56], which solve

$$\min_{x_p \geq 0} \frac{1}{2}\|Ax_p - b_p\|^2 + \gamma\|x_p\|_1 \tag{5.3}$$

for each pixel p. The parameters were chosen so that the l_1, l_1/l_2, and $l_1 - l_2$ approaches all achieved roughly the same level of sparsity, measured as the fraction of nonzero abundances. In particular, for l_1 minimization, we set $\gamma = .08$. The NNLS case corresponds to $\gamma = 0$. For Algorithms 1 and 2 we use a constant but nonzero initialization and set $\epsilon = .001$, $\gamma_0 = .025$, and $C = 10^{-9}\mathrm{I}$. No intrasparsity penalties are used. For Algorithm 1, $\sigma = .1$, $\xi_1 = 2$, and $\xi_2 = 10$, and we stop iterating when the difference in the objective is less than .1. The sparsity and sum of squares errors achieved by the four models are tabulated in Table 2.

The l_1 penalty promotes sparse solutions by trying to move coefficient vectors perpendicular to the positive face of the l_1 ball, shrinking the magnitudes of all elements. The l_1/l_2 penalty and, to some extent, $l_1 - l_2$ promote sparsity by trying to move in a different direction, tangent to the l_2 ball. They do a better job of preserving the magnitudes of the abundances while enforcing a similarly sparse solution. This is reflected in their lower sum of squares errors.

Since the large sum of squares error for l_1 minimization is mostly due to the abundances being too small in magnitude, it doesn't directly indicate which method is better at identifying the support. We therefore recompute the errors after correcting the abundance magnitudes with a debiasing step [21]. We compute the debiased pth column of the abundance matrix by solving an NNLS problem restricted to the estimated support,

$$\min_{x_p \geq 0} \frac{1}{2}\|Ax_p - b_p\|^2 \quad \text{such that} \quad \operatorname{supp}(x_p) \subset \operatorname{supp}(x_p^*), \tag{5.4}$$

where x_p^* denotes the previously estimated vector of abundances. For this experiment, we identify an index i as being in $\operatorname{supp}(x_p^*)$ if $x_p^*(i) > .001$, so $x_p(i) = 0$ whenever $x_p^*(i) \leq .001$. The sparsity and sum of squares errors after debiasing are shown in Table 3. The sum of squares error for l_1 minimization is significantly reduced after correcting the abundance magnitudes, but it remains higher than for l_1/l_2 or $l_1 - l_2$ minimization. This indicates that the support of the abundance matrix is better estimated by l_1/l_2 and $l_1 - l_2$ minimization.

The results of these unmixing algorithms (without debiasing) are also represented in Figure 12 as fraction planes, which are the rows of the abundance matrix visualized as images. They show the spatial abundance of each endmember.

5.4. Hyperspectral unmixing with intra- and intersparsity penalties. In this section we consider a hyperspectral unmixing example with an expanded dictionary consisting of groups

Table 3

Fraction of nonzero abundances and sum of squares error for four unmixing models after debiasing.

	NNLS	l_1	l_1/l_2	$l_1 - l_2$
Fraction nonzero	0.4732	0.2657	0.2639	0.2669
Sum of squares error	1111.2	1369.0	1269.5	1257.4

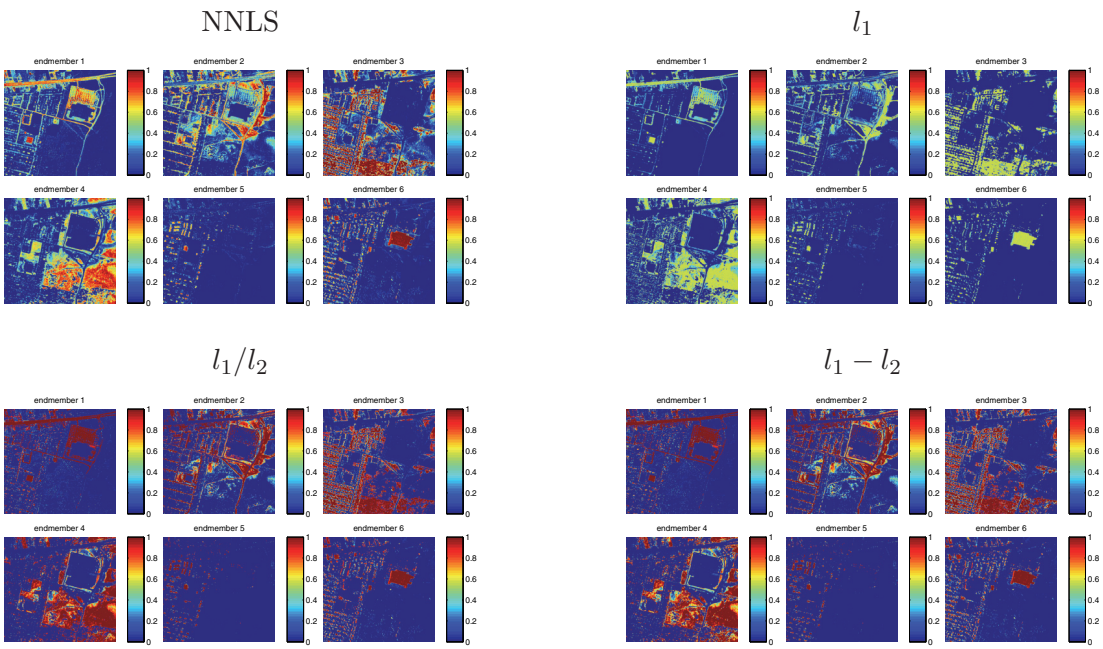

Figure 12. *Estimated fraction planes for urban data using hand selected endmembers.*

of references, each group consisting of candidate endmembers for a particular material. The data we use for this example is from [1] and consists of a 204-band hyperspectral image of crops, soils, and vineyards in Salinas Valley, California. Using a given ground truth labeling, we extract just the data corresponding to romaine lettuce at 4, 5, 6, and 7 weeks, respectively. For each of these four groups, we remove outliers and then randomly extract 100 representative signatures. These and their normalized averages are plotted in Figure 13 and give a sense of the variability of the signatures corresponding to a particular label.

By concatenating the four groups of 100 signatures we construct a dictionary $A_{\text{group}} \in \mathbb{R}^{204\times 400}$. We also construct two smaller dictionaries A_{mean} and $A_{\text{bad}} \in \mathbb{R}^{204\times 4}$. The columns of A_{mean} are the average spectral signatures shown in red in Figure 13, and the columns of A_{bad} are the candidate signatures farthest from the average shown in green in Figure 13.

Synthetic data $b \in \mathbb{R}^{204\times 1560}$ was constructed by randomly constructing a ground truth abundance matrix $\bar{S}_{\text{group}} \in \mathbb{R}^{400\times 1560}$ with 1000 1-sparse columns, 500 2-sparse columns, 50 3-sparse columns, and 10 4-sparse columns, with each group of 100 coefficients being at most 1-sparse. Zero mean Gaussian noise η was also added so that

$$b = A_{\text{group}}\bar{S}_{\text{group}} + \eta.$$

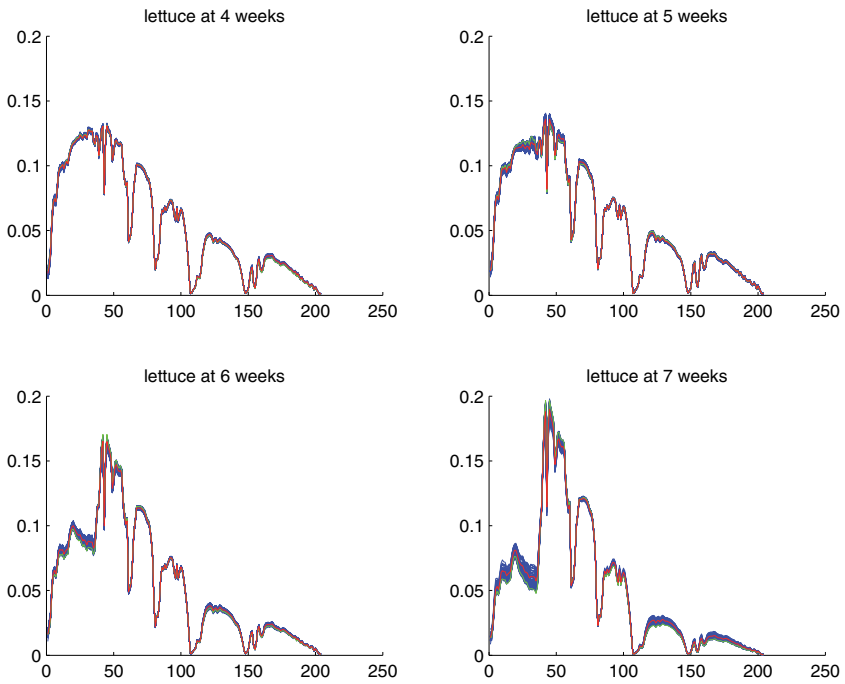

Figure 13. *Candidate endmembers (blue) for romaine lettuce at* $4, 5, 6$, *and* 7 *weeks from Salinas dataset, normalized averages (red), and candidate endmembers farthest from the average (green).*

Each k-sparse abundance column was constructed by first randomly choosing k groups, then randomly choosing one element within each of the selected groups and assigning a random magnitude in $[0, 1]$. The generated columns were then rescaled so that the columns of the noise-free data matrix would have unit l_2 norm.

Define $T \in \mathbb{R}^{4\times 400}$ to be a block diagonal matrix with 1-by-100 row vectors of 1's as the blocks:

$$T = \begin{bmatrix} 1\cdots 1 & & & \\ & 1\cdots 1 & & \\ & & 1\cdots 1 & \\ & & & 1\cdots 1 \end{bmatrix}.$$

Applying T to $\bar{S}_{\text{group}}$ lets us construct a ground truth group abundance matrix $\bar{S} \in \mathbb{R}^{4\times 1560}$ by summing the abundances within groups. For comparison purposes, this will allow us to apply different unmixing methods using the different sized dictionaries A_{mean}, A_{group}, and A_{bad} to compute S_{mean}, TS_{group}, and S_{bad}, respectively, which can then be compared to $\bar{S}$.

We compare six different unmixing methods using the three dictionaries:

1. NNLS (1.1) using A_{mean}, A_{group}, and A_{bad};
2. l_1 (5.3) using A_{mean}, A_{group}, and A_{bad};
3. l_1/l_2 (Problem 1) intersparsity only, using A_{mean} and A_{bad};
4. $l_1 - l_2$ (Problem 2) intersparsity only, using A_{mean} and A_{bad};

5. l_1/l_2 intra- and intersparsity, using A_{group};
6. $l_1 - l_2$ intra- and intersparsity, using A_{group}.

For l_1 unmixing, we set $\gamma = .1$ for A_{mean} and A_{bad} and $\gamma = .001$ for A_{group}. In all applications of Algorithms 1 and 2, we use a constant but nonzero initialization and set $\epsilon_j = .01$, $\gamma_0 = .01$, and $C = 10^{-9}\text{I}$. For the applications with intrasparsity penalties, $\gamma_j = .0001$ for $j = 1, 2, 3, 4$. Otherwise $\gamma_j = 0$. For Algorithm 1, we again use $\sigma = .1$, $\xi_1 = 2$, and $\xi_2 = 10$. We stop iterating when the difference in the objective is less than .001. We repeat these experiments for three different noise levels with standard deviations of .0025, .005, and .01. The corresponding signal-to-noise ratios are 28.94, 22.93, and 16.91, respectively.

We compare the computed group abundances to the ground truth $\bar{S}$ in two ways in Table 4. Measuring the l_0 norm of the difference of abundance matrices indicates how accurately the sparsity pattern was estimated. For each material, we also compute the absolute value of each group abundance error averaged over all measurements. For visualization, we plot the computed number of nonzero entries versus the ground truth for each column of the group abundances in Figure 14.

We see in Table 4 and Figure 14 that NNLS did a poor job of finding sparse solutions although average coefficient errors were low. On the other hand, l_1 minimization did a good job of finding a sparse solution, but coefficient errors were higher because the abundance magnitudes were underestimated. The l_1/l_2 and $l_1 - l_2$ minimization approaches were better at encouraging sparse solutions while maintaining small average errors in the abundance coefficients.

For this example, the average signatures used in A_{mean} turned out to be good choices for the endmembers, and we didn't see any improvement in the estimated group abundances by considering the expanded dictionary A_{group}. However, compared to using the four poorly selected endmember candidates in A_{bad}, we got better results with the expanded dictionary. In the expanded dictionary case, which resulted in an underdetermined dictionary matrix, the abundances S_{group} directly computed by l_1 minimization were much less sparse than those computed by l_1/l_2 and $l_1 - l_2$ minimization. This is because l_1/l_2 and $l_1 - l_2$ minimization were able to enforce 1-sparsity within coefficient groups, but l_1 was not. If the group 1-sparsity requirement is important for the model to be accurate, then this is an advantage of using the l_1/l_2 and $l_1 - l_2$ penalties. Here, this difference in sparsity turned out to not have much effect on the group abundances TS_{group}, which were computed by summing the abundances within each group. This may not hold in situations where the endmember variability is more nonlinear. For example, if the endmember variability had to do with misalignment, as with the earlier DOAS example, then linear combinations of misaligned signatures would not produce a good reference signature.

6. Conclusions and future work. We proposed a method for linear unmixing problems where the dictionary contains multiple references for each material and we want to collaboratively choose the best one for each material present. More generally, we showed how to use l_1/l_2 and $l_1 - l_2$ penalties to obtain structured sparse solutions to nonnegative least squares problems. These were reformulated as constrained minimization problems with differentiable but nonconvex objectives. A scaled gradient projection method based on difference of convex programming was proposed. This approach requires solving a sequence of strongly quadratic programs, and we showed how these can be efficiently solved using the alternating direction

Table 4

Errors between computed group abundance and ground truth $\bar{S}$, where $E_j^{\text{mean}} = \frac{1}{P}\sum_{p=1}^{P}|S_{\text{mean}}(j,p) - \bar{S}(j,p)|$, $E_j^{\text{group}} = \frac{1}{P}\sum_{p=1}^{P}|(TS_{\text{group}})(j,p) - \bar{S}(j,p)|$, *and* $E_j^{\text{bad}} = \frac{1}{P}\sum_{p=1}^{P}|S_{\text{bad}}(j,p) - \bar{S}(j,p)|$.

	SNR = 28.94			
	NNLS	l_1	l_1/l_2	$l_1 - l_2$
$\|S_{\text{mean}} - \bar{S}\|_0$	1488	934	694	776
E_1^{mean}	0.0667	0.1475	0.0468	0.0440
E_2^{mean}	0.0858	0.1580	0.0580	0.0666
E_3^{mean}	0.0607	0.1485	0.0704	0.0930
E_4^{mean}	0.0365	0.1235	0.0418	0.0506
$\|TS_{\text{group}} - \bar{S}\|_0$	1763	819	693	791
E_1^{group}	0.0391	0.1360	0.0530	0.0463
E_2^{group}	0.0604	0.1401	0.0620	0.0720
E_3^{group}	0.0642	0.1496	0.0773	0.1046
E_4^{group}	0.0385	0.1197	0.0469	0.0545
$\|S_{\text{bad}} - \bar{S}\|_0$	2182	1078	957	1048
E_1^{bad}	0.0722	0.1458	0.0490	0.0488
E_2^{bad}	0.1301	0.1432	0.0658	0.0675
E_3^{bad}	0.1143	0.1580	0.0776	0.1077
E_4^{bad}	0.0636	0.1476	0.0551	0.0740
	SNR = 22.93			
	NNLS	l_1	l_1/l_2	$l_1 - l_2$
$\|S_{\text{mean}} - \bar{S}\|_0$	1569	907	770	799
E_1^{mean}	0.0858	0.1526	0.0714	0.0748
E_2^{mean}	0.1044	0.1563	0.0851	0.0884
E_3^{mean}	0.0838	0.1404	0.0764	0.0801
E_4^{mean}	0.0484	0.1169	0.0421	0.0423
$\|TS_{\text{group}} - \bar{S}\|_0$	1764	822	878	831
E_1^{group}	0.0666	0.1427	0.0814	0.0676
E_2^{group}	0.0943	0.1389	0.0937	0.0813
E_3^{group}	0.0988	0.1413	0.1059	0.0997
E_4^{group}	0.0589	0.1118	0.0610	0.0526
$\|S_{\text{bad}} - \bar{S}\|_0$	2126	1078	1072	1029
E_1^{bad}	0.0889	0.1531	0.0707	0.0671
E_2^{bad}	0.1415	0.1431	0.0899	0.0774
E_3^{bad}	0.1305	0.1523	0.0871	0.0922
E_4^{bad}	0.0721	0.1416	0.0568	0.0648
	SNR = 16.91			
	NNLS	l_1	l_1/l_2	$l_1 - l_2$
$\|S_{\text{mean}} - \bar{S}\|_0$	1656	1016	1140	1041
E_1^{mean}	0.1133	0.1583	0.1124	0.1146
E_2^{mean}	0.1339	0.1633	0.1281	0.1229
E_3^{mean}	0.1175	0.1444	0.1197	0.0996
E_4^{mean}	0.0709	0.1248	0.0670	0.0547
$\|TS_{\text{group}} - \bar{S}\|_0$	1839	974	1140	1092
E_1^{group}	0.1038	0.1529	0.1122	0.1018
E_2^{group}	0.1353	0.1529	0.1352	0.1144
E_3^{group}	0.1314	0.1442	0.1320	0.1112
E_4^{group}	0.0819	0.1197	0.0766	0.0640
$\|S_{\text{bad}} - \bar{S}\|_0$	2064	1148	1342	1209
E_1^{bad}	0.1144	0.1557	0.1005	0.0997
E_2^{bad}	0.1589	0.1529	0.1323	0.1137
E_3^{bad}	0.1479	0.1587	0.1221	0.1060
E_4^{bad}	0.0886	0.1512	0.0758	0.0711

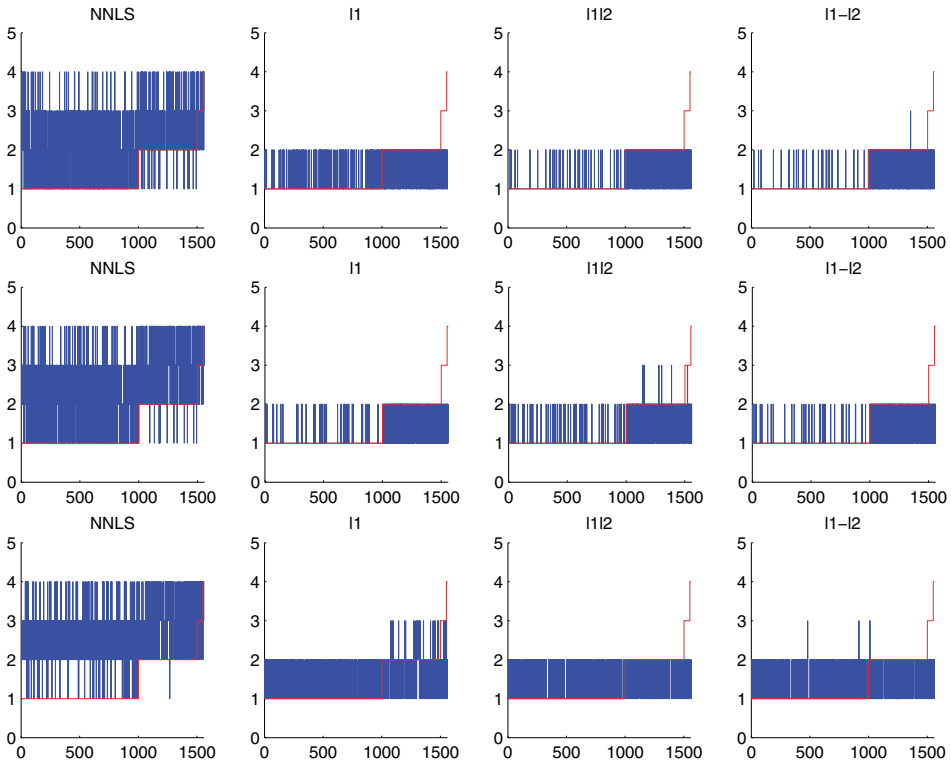

Figure 14. *Estimated number of nonzero entries in each abundance column (blue) and ground truth (red) for the medium noise case, SNR* = 22.93. *Row* 1: S_{mean}. *Row* 2: TS_{group}. *Row* 3: S_{bad}.

method of multipliers. Moreover, few iterations were required in practice, between 4 and 20 for all of the numerical examples presented in this paper. Some convergence analysis was also presented to show that limit points of the iterates are stationary points. Numerical results for unmixing problems in differential optical absorption spectroscopy and hyperspectral image analysis show that our difference of convex approach using l_1/l_2 and $l_1 - l_2$ penalties is capable of promoting different levels of sparsity on possibly overlapping subsets of the fitting or abundance coefficients.

In future work we would like to test this method on more general multiple choice quadratic knapsack problems, which are related to the applications presented here that focused on finding solutions that were at most 1-sparse within specified groups. It would be interesting to see how this variational approach performs relative to combinatorial optimization strategies for similar problems. We are also interested in exploring alternative sparsity penalties that can be adapted to the dataset. When promoting 1-sparse solutions, the experiments in this paper used fixed sparsity parameters that were simply chosen to be sufficiently large. We are interested in justifying the technique of gradually increasing this parameter while iterating, which empirically seems better able to avoid bad local minima. The applications presented here all involved uncertainty in the dictionary, which was expanded to include multiple candidate references for each material. If a priori assumptions are available about the relative likelihood of these candidates, we would like to incorporate this into the model.

Acknowledgments. The authors would like to thank Professors Russ Caflisch, Ingrid Daubechies, Tom Hou, and Stan Osher for their interest and the opportunity to present some of the results at the Adaptive Data Analysis and Sparsity Workshop at the Institute for Pure and Applied Mathematics at UCLA, Jan. 31, 2013. We thank Lisa Wingen for providing DOAS references and data, which we used as a guide when generating synthetic data for some of our numerical examples. Thanks to John Greer for pointing out a paper by Zare and Gader [67]. We would also like to thank the anonymous referees for their constructive comments.

REFERENCES

[1] *AVIRIS Salinas Valley Dataset*, http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.
[2] *Urban Dataset*, US Army Topographic Engineering Center, http://www.tec.army.mil/hypercube.
[3] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, *Model-based compressive sensing*, IEEE Trans. Inform. Theory, 56 (2010), pp. 1982–2001.
[4] M. Berry, M. Browne, A. Langville, P. Pauca, and R. J. Plemmons, *Algorithms and applications for approximate nonnegative matrix factorization*, Comput. Statist. Data Anal., 52 (2007), pp. 155–173.
[5] A. Bertozzi, S. Esedoḡlu, and A. Gillette, *Analysis of a two-scale Cahn–Hilliard model for binary image inpainting*, Multiscale Model. Simul., 6 (2007), pp. 913–936.
[6] D. Bertsekas, *Nonlinear Programming*, Athena Scientific, Nashua, NH, 1999.
[7] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation*, Prentice Hall, Englewood Cliffs, NJ, 1989.
[8] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, *Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches*, IEEE Trans. Image Process., 5 (2012), pp. 354–379.
[9] S. Bonettini, R. Zanella, and L. Zanni, *A scaled gradient projection method for constrained image deblurring*, Inverse Problems, 25 (2009), 015002.
[10] P. Brucker, *An $O(n)$ algorithm for quadratic knapsack problems*, Oper. Res. Lett., 3 (1984), pp. 163–166.
[11] A. M. Bruckstein, M. Elad, and M. Zibulevsky, *On the uniqueness of nonnegative sparse solutions to underdetermined systems of equations*, IEEE Tran. Inform. Theory, 54 (2008), pp. 4813–4820.
[12] E. Candes, J. Romberg, and T. Tao, *Stable signal recovery from incomplete and inaccurate measurements*, Comm. Pure Appl. Math., 59 (2006), pp. 1207–1223.
[13] A. Castrodad, Z. Xing, J. B. Greer, E. Bosch, L. Carin, and G. Sapiro, *Learning discriminative sparse representations for modeling, source separation, and mapping of hyperspectral imagery*, IEEE Trans. Geosci. Remote Sens., 49 (2011), pp. 4263–4281.
[14] A. Chambolle and T. Pock, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision, 40 (2011), pp. 120–145.
[15] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.
[16] O. Eches, N. Dobigeon, C. Mailhes, and J.-Y. Tourneret, *Bayesian estimation of linear mixtures using the normal compositional model. Application to hyperspectral imagery*, IEEE Trans. Image Process., 19 (2010), pp. 1403–1413.
[17] M. T. Eismann and D. Stein, *Stochastic mixture modeling*, in Hyperspectral Data Exploitation: Theory and Applications, C.-I Chang, ed., John Wiley & Sons, Hoboken, NJ, 2007, pp. 107–148.
[18] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, *A convex model for nonnegative matrix factorization and dimensionality reduction on physical space*, IEEE Trans. Image Process., 21 (2012), pp. 3239–3252.
[19] E. Esser, X. Zhang, and T. F. Chan, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci., 3 (2010), pp. 1015–1046.
[20] D. J. Eyre, *An Unconditionally Stable One-Step Scheme for Gradient Systems*, Technical report, De-

partment of Mathematics, University of Utah, Salt Lake City, UT, 1998; available online from www.math.utah.edu/~eyre/research/methods/stable.ps.

[21] M. Figueiredo, R. Nowak, and S. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems*, IEEE J. Sel. Topics Signal Process., 1 (2007), pp. 586–597.

[22] B. J. Finlayson-Pitts, *unpublished data*, provided by L. Wingen, Department of Chemistry, University of California at Irvine, Irvine, CA, 2000.

[23] J. Friedman, T. Hastie, and R. Tibshirani, *A Note on the Group Lasso and a Sparse Group Lasso*, Technical report, Department of Statistics, Stanford University, Stanford, CA, 2010.

[24] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.

[25] R. Glowinski and A. Marrocco, *Sur l'approximation par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problemès de Dirichlet non linéaires*, RAIRO Anal. Numér., 9 (1975), pp. 41–76.

[26] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.

[27] J. Greer, *Sparse demixing*, in Proc. SPIE 7695, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVI, 2010, 76951O.

[28] Z. Guo and S. Osher, *Template matching via l_1 minimization and its application to hyperspectral data*, Inverse Probl. Imaging, 5 (2011), pp. 19–35.

[29] Z. Guo, T. Wittman, and S. Osher, *L1 unmixing and its application to hyperspectral image enhancement*, in Proc. SPIE 7334, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV, 2008, 73341M.

[30] P. O. Hoyer, *Non-negative sparse coding*, in Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 557–565.

[31] P. O. Hoyer, *Non-negative matrix factorization with sparseness constraints*, J. Mach. Learn. Res., 5 (2003/04), pp. 1457–1469.

[32] Y. H. Hu, H. B. Lee, and F. L. Scarpace, *Optimal linear spectral unmixing*, IEEE Trans. Geosci. Remote Sens., 37 (1999), pp. 639–644.

[33] J. Huang, T. Zhang, and D. Metaxas, *Learning with structured sparsity*, J. Mach. Learn. Res., 12 (2011), pp. 3371–3412.

[34] N. Hurley and S. Rickard, *Comparing measures of sparsity*, IEEE Trans. Inform. Theory, 55 (2009), pp. 4723–4741.

[35] M. D. Iordache, J. M. Bioucas-Dias, and A. Plaza, *Sparse unmixing of hyperspectral data*, IEEE Trans. Geosci. Remote Sens., 49 (2011), pp. 2014–2039.

[36] R. Jenatton, J.-Y. Audibert, and F. Bach, *Structured variable selection with sparsity-inducing norms*, J. Mach. Learn. Res., 12 (2011), pp. 2777–2824.

[37] R. Jenatton, G. Obozinski, and F. Bach, *Structured Sparse Principal Component Analysis*, preprint, arXiv:0909.1440v1 [stat.ML], 2009.

[38] H. Ji, J. Li, Z. Shen, and K. Wang, *Image deconvolution using a characterization of sharp images in wavelet domain*, Appl. Comput. Harmon. Anal., 32 (2012), pp. 295–304.

[39] N. Keshava and J. F. Mustard, *Spectral unmixing*, IEEE Signal Processing Mag., 19 (2002), pp. 44–57.

[40] D. Krishnan, T. Tay, and R. Fergus, *Blind deconvolution using a normalized sparsity measure*, in Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 233–240.

[41] K. Lange, D. Hunter, and I. Yang, *Optimization transfer using surrogate objective functions*, J. Comput. Graph. Statist., 9 (2000), pp. 1–20.

[42] C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, NJ, 1974.

[43] D. D. Lee and H. S. Seung, *Algorithms for non-negative matrix factorization*, in Proceedings of NIPS, Advances in Neural Information Processing 13, MIT Press, Cambridge, MA, 2001, pp. 556–562.

[44] Y. Lou, A. L. Bertozzi, and S. Soatto, *Direct sparse deblurring*, J. Math. Imaging Vision, 39 (2011), pp. 1–12.

[45] Z. Lu and Y. Zhang, *Penalty Decomposition Methods for L0-Norm Minimization*, preprint, arXiv:1008.5372v2 [math. OC], 2012.

[46] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, *Online dictionary learning for sparse coding*, in Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09), ACM, New York, 2009, pp. 689–696.
[47] L. Meier, S. van de Geer, and P. Bühlmann, *The group Lasso for logistic regression*, J. R. Stat. Soc. Ser. B Stat. Methodol., 70 (2008), pp. 53–71.
[48] J. M. P. Nascimento and J. M. Bioucas-Dias, *Vertex component analysis: A fast algorithm to unmix hyperspectral data*, IEEE Trans. Geosci. Remote Sens., 43 (2004), pp. 898–910.
[49] V. P. Pauca, J. Piper, and R. J. Plemmons, *Nonnegative matrix factorization for spectral data analysis*, Linear Algebra Appl., 416 (2006), pp. 29–47.
[50] U. Platt and J. Stutz, *Differential Optical Absorption Spectroscopy: Principles and Applications*, Springer, Berlin, Heidelberg, 2008.
[51] Z. Qin and D. Goldfarb, *Structured sparsity via alternating direction methods*, J. Mach. Learn. Res., 13 (2012), pp. 1435–1468.
[52] N. Saito and J-F. Remy, *The polyharmonic local sine transform: A new tool for local image analysis and synthesis without edge effect*, Appl. Comput. Harmon. Anal., 20 (2006), pp. 41–73.
[53] M. Slawski and M. Hein, *Sparse recovery by thresholded non-negative least squares*, in Proceedings of NIPS, Advances in Neural Information Processing 24, MIT Press, Cambridge, MA, 2011, pp. 1926–1934.
[54] B. Sriperumbudur and G. Lanckriet, *On the convergence of the concave-convex procedure*, in Proceedings of NIPS, Advances in Neural Information Processing 22, MIT Press, Cambridge, MA, 2009, pp. 1759–1767.
[55] D. Stein, *Application of the normal compositional model to the analysis of hyperspectral imagery*, in Proceedings of the IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, 2003, pp. 44–51.
[56] A. Szlam, Z. Guo, and S. Osher, *A split Bregman method for non-negative sparsity penalized least squares with applications to hyperspectral demixing*, in Proceedings of the 17th IEEE International Conference on Image Processing (ICIP), 2010, pp. 1917–1920.
[57] P. D. Tao and L. T. H. An, *Convex analysis approach to d.c. programming: Theory, algorithms and applications*, Acta Math. Vietnam., 22 (1997), pp. 289–355.
[58] P. D. Tao and L. T. H. An, *A D.C. optimization algorithm for solving the trust-region subproblem*, SIAM J. Optim., 8 (1998), pp. 476–505.
[59] R. Tibshirani, *Regression shrinkage and selection via the lasso*, J. Roy. Statist. Soc. Ser. B, 58 (1996), pp. 267–288.
[60] J. A. Tropp, *Greed is good: Algorithmic results for sparse approximation*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2231–2242.
[61] B. P. Vollmayr-Lee and A. D. Rutenberg, *Fast and accurate coarsening simulation with an unconditionally stable time step*, Phys. Rev. E, 68 (2003), 066703.
[62] M. E. Winter, *N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data*, in Proc. SPIE 3753, Imaging Spectrometry V, 1999, pp. 266–275.
[63] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.
[64] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, J. Roy. Stat. Soc. Ser. B Stat. Methodol., 68 (2006), pp. 49–67.
[65] A. Yuille and A. Rangarajan, *The concave-convex procedure*, Neural Comput., 15 (2003), pp. 915–936.
[66] A. Zare, *Hyperspectral Endmember Detection and Band Selection Using Bayesian Methods*, Ph.D. thesis, Computer Science Department, University of Florida, Gainesville, FL, 2008; available online at http://gradworks.umi.com/33/47/3347194.html.
[67] A. Zare and P. Gader, *PCE: Piece-wise convex endmember detection*, IEEE Trans. Geosci. Remote Sens., 48 (2010), pp. 2620–2632.
[68] Y. Zhou, R. Jin, and S. C. Hoi, *Exclusive lasso for multi-task feature selection*, in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), JMLR: Workshop and Conference Proceedings, Vol. 9, 2010, pp. 988–995.