Sparse linear filters for detection and classification in hyperspectral imagery

James Theiler[a] and Karen Glocer[a,b]

[a]Space and Remote Sensing Sciences Group,
Los Alamos National Laboratory, Los Alamos, NM, USA;

[b]Department of Computer Science,
University of California Santa Cruz, Santa Cruz, CA, USA.

ABSTRACT

We investigate the use of convex optimization to identify sparse linear filters in hyperspectral imagery. A linear filter is sparse if a large fraction of its coefficients are zero. A sparse linear filter can be advantageous because it only needs to access a subset of the available spectral channels, and it can be applied to high-dimensional data more cheaply than a standard linear detector. Finding good sparse filters is nontrivial because there is a combinatorially large number of discrete possibilities from which to choose the optimal subset of nonzero coefficients. But, by converting the optimality criterion into a convex loss function, and by employing an L1 penalty, one can obtain sparse solutions that are globally optimal. We investigate the performance of these sparse filters as a function of their sparsity, and compare the convex optimization approach with more traditional alternatives for feature selection. The methodology is applied both to the adaptive matched filter for weak signal detection, and to the Fisher linear discriminant for terrain categorization.

Keywords: matched filter, hyperspectral imagery, adaptive signal detection

1. INTRODUCTION

Hyperspectral data provides imagery with hundreds of spectral channels, and this high resolution provides great flexibility and, in some cases, exquisite discrimination ability. But when looking for a specific target, discrimination performance can often remain high, even with a small but carefully chosen subset of the available hyperspectral channels. When only a few spectral channels are used, this enables a reduction in the size of the data set that is necessary and a reduction in the computation required to process that data; having fewer channels in the data also means there are fewer free parameters in the model, and overfitting effects (such as the notorious Hughes effect[1]) are reduced. We also remark that if fewer channels are actually needed, then adaptive sensors may be able to exploit this and more effective acquisition of the data may be possible.

The usefulness of reducing the number of channels for a given application has led to a number of investigations that specifically address the problem in the context of hyperspectral imagery.[2–7] But the problem of finding optimal subsets of variables in high-dimensional data is of broad interest in a variety of fields, and it is of considerable interest to the machine learning community.[8] Our approach here is to adapt this machine-learning approach for hyperspectral remote sensing data; this will provide both specific tools, such as algorithms for convex optimization,[9] and a general point of view that considers the distinction between in-sample and out-of-sample modeling.

2. MATCHED FILTER AND FISHER DISCRIMINANT

We consider two uses for linear discriminators: the detection of weak signals (such as thin gaseous plumes) and the classification of pixels (*e.g.*, for terrain categorization). In both cases, a linear function $\mathcal{D}(\mathbf{r}) = \mathbf{q}^T\mathbf{r}$ is produced, where $\mathbf{r} \in \mathbb{R}^d$ is a vector of spectral components of a pixel, and $\mathbf{q} \in \mathbb{R}^d$ is a vector of coefficients. The scalar-valued $\mathcal{D}(\mathbf{r})$ is typically compared against a threshold q_o to decide if the signal is detected at the pixel, or to predict which of two classes is represented by the pixel.

Email: {jt,glocer}@lanl.gov

2.1. Gaseous plume and adaptive matched filter

In the weak plume limit, the effect of the plume can be represented as a superposition of signal and background

$$\mathbf{r} = \epsilon\mathbf{b} + \mathbf{z} \tag{1}$$

where $\mathbf{r}$ is the observed radiance, $\mathbf{b}$ is the *known* plume signature, and $\mathbf{z}$ is the *unknown* background clutter. Eq. (1) corresponds to the radiance in a single pixel; here, $\mathbf{r}$, $\mathbf{b}$, and $\mathbf{z}$ are d-dimensional vectors corresponding to the d channels of the hyperspectral image, and ϵ is a scalar corresponding to the plume strength.

The signal is proportional to $\mathbf{q}^T\mathbf{b}$, and the rms of the clutter is $\left\langle\, |\mathbf{q}^T(\mathbf{z} - \langle\, \mathbf{z}\,\rangle)|^2 \,\right\rangle^{1/2}$, where the angle brackets $\langle\,\cdot\,\rangle$ indicate average over the pixels in the scene. With the covariance matrix defined as

$$K = \left\langle\, (\mathbf{z} - \langle\mathbf{z}\rangle)(\mathbf{z} - \langle\mathbf{z}\rangle)^T \,\right\rangle, \tag{2}$$

the signal-to-clutter ratio is given by

$$\mathtt{SCR}(\mathbf{q}) = \frac{\epsilon\, \mathbf{q}^T\mathbf{b}}{\sqrt{\mathbf{q}^T K \mathbf{q}}}. \tag{3}$$

The adaptive matched filter (sometimes called "clutter matched filter" or simply "matched filter") is the vector $\mathbf{q}$ that maximizes the SCR. It bears remarking that the signal-to-clutter ratio is entirely independent of the magnitude of $\mathbf{q}$. It is the *direction* of the matched filter $\mathbf{q}$, and not its magnitude, that matters for SCR optimization.

The optimization of Eq. (3) is equivalent to

$$\mathbf{q}_{\text{AMF}} = \underset{\mathbf{q}}{\text{argmax}}\ \mathbf{q}^T\mathbf{b} \quad \text{subject to} \quad \mathbf{q}^T K \mathbf{q} = 1. \tag{4}$$

which leads to the Lagrangian formulation: $\mathbf{q}_{\text{AMF}}$ minimizes

$$\mathcal{L}_\mu(\mathbf{q}; \mathbf{b}, K) = -\mathbf{q}^T\mathbf{b} + \mu\mathbf{q}^T K \mathbf{q} \tag{5}$$

for some Lagrange multiplier $\mu > 0$. This is algebraically equivalent to

$$\mathcal{L}_\mu(\mathbf{q}; \mathbf{b}, K) = \frac{1}{4\mu}\left[(2\mu K\mathbf{q} - \mathbf{b})^T K^{-1}(2\mu K\mathbf{q} - \mathbf{b}) - \mathbf{b}^T K^{-1}\mathbf{b}\right], \tag{6}$$

and it is clear that the optimal solution is given when $2\mu K\mathbf{q} - \mathbf{b} = 0$. If we take the $\mathbf{q}^T K\mathbf{q} = 1$ constraint literally, then this leads to $\mu = \frac{1}{2}\sqrt{\mathbf{b}^T K^{-1}\mathbf{b}}$. But since μ only sets the magnitude of $\mathbf{q}$, and since that magnitude does not affect the SCR, we will henceforth take $\mu = \frac{1}{2}$. This leads to

$$\mathcal{L}(\mathbf{q}; \mathbf{b}, K) = -\mathbf{q}^T\mathbf{b} + \tfrac{1}{2}\mathbf{q}^T K \mathbf{q} \tag{7}$$

for which the optimal solution is $\mathbf{q}_{\text{AMF}} = K^{-1}\mathbf{b}$.

2.2. Binary classification and the Fisher linear discriminant

Given data $(\mathbf{x}_i, y_i)$ with $\mathbf{x}_i \in R^d$ representing measurements that are available, and $y_i \in \{-1, 1\}$ denoting a label associated with the data sample, the aim of a classification algorithm is to find a function $f(\mathbf{x})$ such that $\text{sign}(f(\mathbf{x}_i))$ predicts the label y_i.

In the applying the Fisher discriminant to this problem, it is assumed that the data points $\mathbf{x}_i$ associated with $y_i = -1$ are distributed as a gaussian with mean $\boldsymbol{\mu}_-$ and covariance K. The data points in the other class, the measurements $\mathbf{x}_i$ associated with $y_i = 1$ are distributed as a gaussian with a different mean, $\boldsymbol{\mu}_+$ but with the same covariance K. In practice $\boldsymbol{\mu}_\pm$ are estimated from the sample means of the data in the two classes, and K is estimated as a pooled covariance from all of the data points. The Fisher discriminant is given by[10] $f(\mathbf{x}) = \mathbf{q}^T\mathbf{x} + q_o$ where

$$\mathbf{q} = K^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-). \tag{8}$$

The threshold is traditionally given by $q_o = \frac{1}{2}\mathbf{q}^T(\boldsymbol{\mu}_+ + \boldsymbol{\mu}_-)$, but in practice one can adjust q_o to trade off false alarms and missed detections.

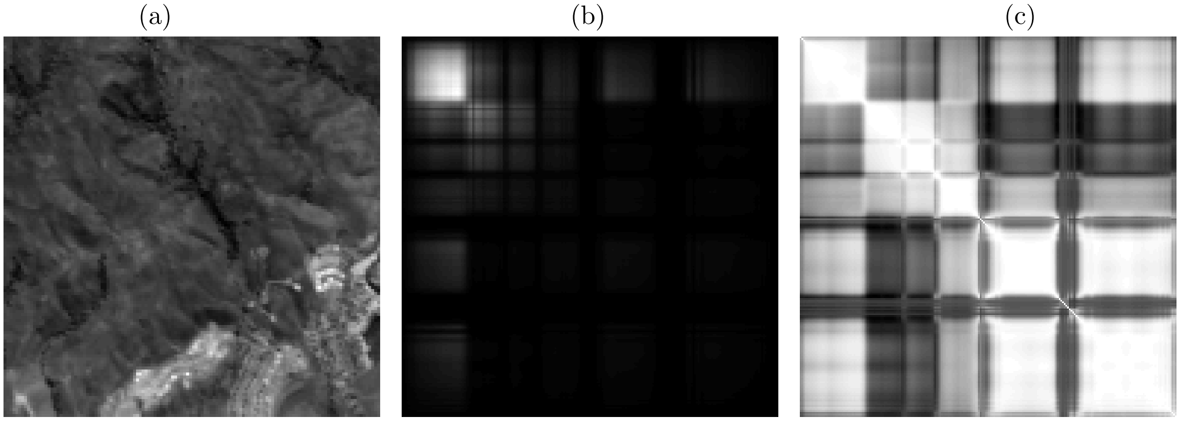

Figure 1. **(a)** Broadband image (sum of the 224 channels) of AVIRIS scene used to compute covariance matrix K. **(b)** Covariance matrix displayed as an image, where brightness corresponds to the value of the matrix elements. **(c)** Diagonal-normalized covariance matrix $K_o = D^{-1/2}KD^{-1/2}$, where D is a diagonal matrix with the same diagonal elements as the diagonal elements of the matrix K.

2.3. Sparse filter optimization

Let $\mathcal{A}$ be the set of indices of the vector $\mathbf{q}$ for which at component is nonzero. In general, for an optimal filter, we do not expect *any* components of $\mathbf{q}$ to be strictly zero, and in that case $\mathcal{A} = \{1, \ldots, d\}$. But we can restrict our attention to vectors $\mathbf{q}$ for which $q_i = 0$ when $i \notin \mathcal{A}$. In general, when we write $\mathbf{q}_\mathcal{A}$, we are referring to a truncated vector which includes only the elements q_i where $i \in \mathcal{A}$. We will write

$$\mathcal{L}_\mathcal{A}(\mathbf{q}; K, \mathbf{b}) = -\mathbf{q}_\mathcal{A}^T \mathbf{b}_\mathcal{A} + \tfrac{1}{2}\mathbf{q}_\mathcal{A}^T K_{\mathcal{A}\mathcal{A}} \mathbf{q}_\mathcal{A} \tag{9}$$

and note that this is optimized when $\mathbf{q}_\mathcal{A} = K_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{b}_\mathcal{A}$, and that in that case

$$\mathcal{L}_\mathcal{A} = -\tfrac{1}{2}\mathbf{b}_\mathcal{A}^T K_{\mathcal{A}\mathcal{A}}^{-1} \mathbf{b}_\mathcal{A}. \tag{10}$$

Once we are given $\mathcal{A}$, then it is straightforward to find the optimal $\mathbf{q}$ subject to $q_i = 0$ for $i \notin \mathcal{A}$.

But the core of the sparse matched filter problem is to find an appropriate $\mathcal{A}$. There are 2^d candidates, and this is, in general, an NP-hard problem. The problem is often expressed is in terms of $|\mathcal{A}|$, the number of nonzero components:

$$\text{minimize } \left[-\mathbf{q}^T\mathbf{b} + \tfrac{1}{2}\mathbf{q}^T K \mathbf{q}\right] \quad \text{subject to} \quad q_i = 0 \text{ for } i \notin \mathcal{A} \quad \text{and} \quad |\mathcal{A}| \leq n. \tag{11}$$

Or, in terms of Eq. (10), we can write

$$\text{maximize } \mathbf{b}_\mathcal{A}^T K_{\mathcal{A}\mathcal{A}}^{-1} \mathbf{b}_\mathcal{A} \quad \text{subject to} \quad |\mathcal{A}| \leq n. \tag{12}$$

2.4. Out-of-sample performance

From a training set of data, we can compute a covariance K_{train}, and from that produce a matched filter $\mathbf{q}$ (sparse or otherwise) that optimizes the SCR given by $\mathbf{q}^T\mathbf{b}/\sqrt{\mathbf{q}^T K_{\text{train}}\mathbf{q}}$. However, although it may perform well against this in-sample covariance, that does not guarantee that it will perform well out of sample, on a test set of data. That is: given K_{train}, computed from pixels drawn from the same distribution (or in this case, the same image) as the training set, how good is the out-of-sample SCR, given by $\mathbf{q}^T\mathbf{b}/\sqrt{\mathbf{q}^T K_{\text{test}}\mathbf{q}}$.

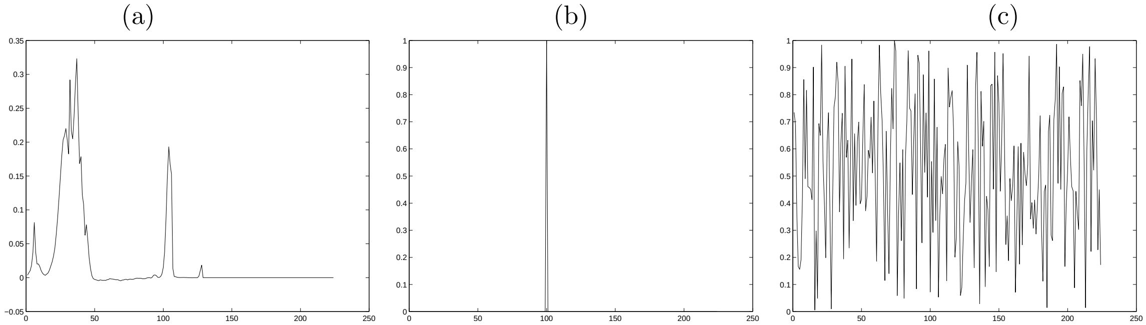

Figure 2. (a) Artificial signature vector **b**, derived from a freon spectrum, but shifted from the infrared to the visible. **(b)** Truly artificial signature vector is zero in all channels except for channel 100. **(c)** Positive random signature.

3. DATA

Experiments with sparse matched filters requires only a covariance matrix K, and a signature vector $\mathbf{b}$. To get a realistic covariance matrix, we will use 128x128 chip of 224-channel AVIRIS data.[11] This chip is from the `f97062t01p02_r03` data set (Moffet Field), and is available from the AVIRIS website.[12] The image is shown in Fig. 1(a), and the elements of the covariance matrix are shown as in image in Fig. 1(b).

For our numerical experiments, we will consider the three different signature vectors shown in Fig. 2. The first signature is based on a freon absorption spectrum;[*] the second signature is maximally sparse in its own right, it is a single spike at channel 100; while the third signature is very dense, with every channel a positive random number.

3.1. Diagonal normalization

For a given covariance matrix K and signal b, we will occasionally consider a diagonal-normalized version. If D is any positive diagonal matrix, then the transformation

$$K_o = D^{-1/2} K D^{-1/2} \tag{13}$$

$$\mathbf{b}_o = D^{-1/2}\mathbf{b} \tag{14}$$

leads to a $K_o, \mathbf{b}_o$ which is equivalent to $K, \mathbf{b}$ when it comes to the sparse optimization problem. This is because if $\mathbf{q}$ optimizes $\mathcal{L}(\mathbf{q}; \mathbf{b}, K)$ subject to $q_i = 0$ for $i \notin \mathcal{A}$; then $\mathbf{q}^* = D^{1/2}\mathbf{q}$ optimizes $\mathcal{L}(\mathbf{q}; \mathbf{b}_o, K_o)$ and $q_i^* = (D^{1/2}\mathbf{q})_i = D_{ii}^{1/2} q_i$, so $q_i^* = 0$ for $i \notin \mathcal{A}$. We will take D to have the same diagonal elements as K (that is: $D_{ii} = K_{ii}$, but $D_{ij} = 0$ for $i \neq j$). Then the diagonal elements of K_o will all be unity; this puts all the components i on equal footing. Fig. 1(c) shows the effect of this normalization on the covariance matrix K obtained from the AVIRIS data.

4. ALGORITHMS

4.1. Sequential selection algorithms

Fig. 3(a-e) shows the performance of several sequential feature selection algorithms, applied to the sparse linear filter for detecting the freon signature in Fig. 2(a). The most straightforward of these is stepwise forward selection (SFS). At each step, the feature that most (greedily) improves the SCR is added to the list of features. This is not unduly expensive but it does require an SCR computation for each available feature.

Formally speaking, let use write $\mathcal{A}_t$ as the subset that has been selected at the nth step; since this algorithm simply adds one new feature at each step, we have that $|\mathcal{A}_t| = t$. To find the feature that is added at step $t+1$, we choose

$$j^* = \underset{j}{\operatorname{argmin}}\, \mathcal{L}_{\mathcal{A}_t \cup \{j\}} \tag{15}$$

[*] Freon absorbs in the long-wave infrared, but AVIRIS takes imagery in the visible and near-infrared; the spectrum in Fig. 2(a) has the same "shape" as the long-wave freon spectrum, but has been artificially transplanted to the visible.

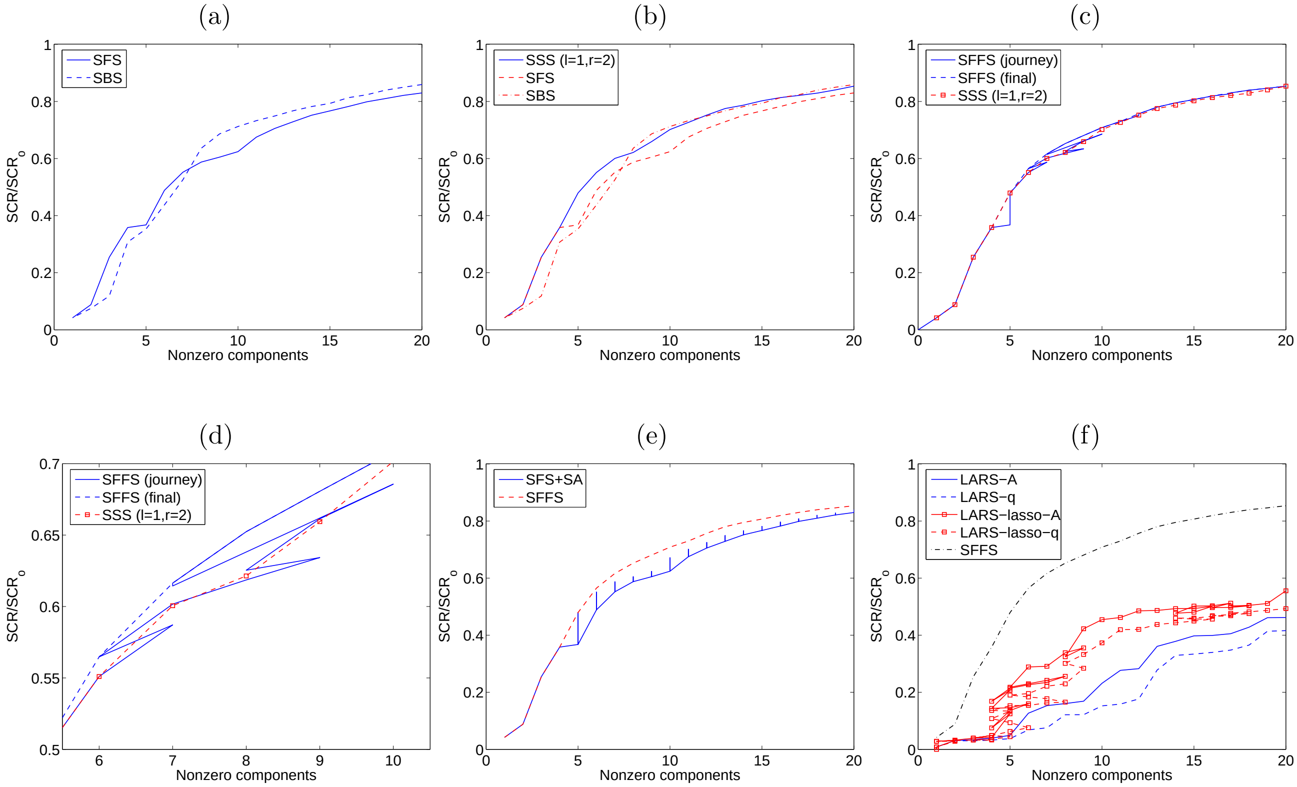

Figure 3. Fraction of SCR plotted against the number of nonzero channels, for the freon signature shown in Fig. 2(a), using the covariance shown in Fig. 1(b). **(a)** Stepwise forward selection (SFS) does slightly better than stepwise backward selection (SBS) for *very* sparse filters, but SBS does better for larger number of nonzero coefficients. **(b)** Stearn's stepwise selection (SSS) which takes r forward steps, followed by l steps back, does better than SFS and generally better than SBS. **(c)** The stepwise floating forward selection (SFFS) algorithm involves a scheme in which each iteration takes a single forward step and then follows that with conditional backward steps. The solid line indicates the "journey" that SFFS takes as it optimizes the feature selection. The dashed line is the final result obtained by SFFS. For comparison SSS is shown, which does not quite do as well. **(d)** This is just a close-up of (c); showing the circuitous journey that SFFS takes. **(e)** SFS+SA includes a steepest ascent (SA) improvement of the SFS solution, and these are shown as vertical spines on the SFS curve. SFFS is still better. **(f)** Comparison of four variants of LARS, described in the text, with SFFS, shows that for these convex optimization methods, collectively, perform substantially worse than heuristic sequential selection algorithms.

and then $\mathcal{A}_{t+1} = \mathcal{A}_t \cup \{j^*\}$.

A simple variant of SFS is sequential backward selection (SBS). Here, the starting point is to have all features selected and to delete one feature at at time, based on which deletion gives the best performance. As Fig. 3(a) shows, SFS performs a little better than SBS for very small numbers of features, whereas SBS is better for larger numbers of features. Both of these have the "nesting property" which means that $|\mathcal{A}_s| < |\mathcal{A}_t|$ implies $\mathcal{A}_s \subset \mathcal{A}_t$.

Stearns[2, 13] suggested a feature selection scheme in which forward and backward selection are alternated. In this (l, r) scheme, r forward selection steps are taken followed by l backward selection steps. This avoids the nesting constraint, and combines the advantages of SFS and SBS. While it is not clear how to choose the optimal values of the parameters l and r, even simple choices like $r = 2$ and $l = 1$ can lead to improvements. Pudil *et al.*[14] proposed a modification of this idea, called stepwise floating forward selection (SFFS), which takes one forward step, and then takes an adaptive number of backward steps. This has proven to be an effective heuristic – although it cannot guarantee the absolute optimal solution, numerical comparisons by Jain and Zongker[15] favor the floating stepwise approach. Our own comparisons in Fig. 3(c) confirm this observation.

Serpico and Bruzzone[3] described two local improvement approaches, when you start with a subset of k

```
1. Initialize: $j_o = \mathrm{argmax}\,|b_j|$; $\mathcal{A} = \{j_o\}$; $\mathbf{q} = 0$
2. $\hat{\mathbf{b}} = \mathbf{b} - K\mathbf{q}$
3. $\mathbf{s}_{\mathcal{A}} = \mathrm{sign}(\hat{\mathbf{b}}_{\mathcal{A}})$
4. $\mathbf{w}_{\mathcal{A}} = K_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{s}_{\mathcal{A}}$
5. We want $\mathbf{q} \leftarrow \mathbf{q} + \gamma\mathbf{w}$, but need the appropriate value of $\gamma > 0$.
    • Let $\mathbf{c}(\gamma) = \mathbf{b} - K(\mathbf{q} + \gamma\mathbf{w}) = \hat{\mathbf{b}} - \gamma K\mathbf{w}$ and note that for every component $j \in \mathcal{A}$, we have $c_j(\gamma) = \hat{b}_j - \gamma(K\mathbf{w})_j = \hat{b}_j - \gamma s_j$; that is, the $|c_j(\gamma)|$ are equal for all $j \in \mathcal{A}$ and all $\gamma$.
    • Since they are all equal, let $c_{\mathcal{A}}(\gamma) = |c_j(\gamma)|$ for some $j \in \mathcal{A}$.
    • For $\gamma = 0$, we know that $c_{\mathcal{A}}(\gamma) > |c_k(\gamma)|$ for all $k \notin \mathcal{A}$, so by continuity, we expect that $c_{\mathcal{A}}(\gamma) > |c_k(\gamma)|$ for small enough $\gamma$.
    • The idea is to find the smallest $\gamma$ for which the above condition fails. That is find $k \notin \mathcal{A}$ that has the smallest positive value of $\gamma$ that solves $c_{\mathcal{A}}(\gamma) = |c_k(\gamma)|$.
    • $\mathbf{q} = \mathbf{q} + \gamma\mathbf{w}$
6. $\mathcal{A} = \mathcal{A} \cup \{k\}$
7. Go to 2
```

Figure 4. Pseudocode for applying LARS to the sparse matched filter problem. At each iteration, a new feature k is added, and the sparse matched filter $\mathbf{q}$ is updated. The LARS-lasso algorithm includes an extra test, not shown, testing whether there is a value of γ for which one of the features in $\mathcal{A}$ should be removed from $\mathcal{A}$, according to the lasso criterion.

selected features, and you seek a better subset by swapping out selected features with unselected features. The straightforward and expensive approach (SA = steepest ascent) considers all possibilities, but a shortcut (FCS = fast constrained search) is not as expensive and does nearly as well. Further comparisons of these methods are described by Serpico *et. al.*[4] However, our own comparisons in Fig. 3(d) indicate that even the expensive SA (whose performance will upper-bound that of FCS) method does not do as well as the SFFS algorithm.

4.2. Convex optimization algorithms

The problem with the various sequential selection algorithms is that they are trying to solve an NP-hard problem. They are trying to optimize a loss function with many local minima. Although various branch-and-bound approaches have been suggested[16–18] to substantially reduce the computation, there are still exponentially many band combination candidates that must be evaluated if the optimal solution is to be found. Heuristics such as SFS and SFFS are found, in many cases (including in the cases we looked at), to produce reasonable solutions, but it is difficult to provide theoretical assurance that adequate performance will always be achieved.

An entirely different approach involves finding *the* global optimum of a loss function which only *approximates* the actual loss function of interest. In other words, instead of using a heuristic algorithm to find a suboptimal solution, an exact algorithm will be used to fine the precise optimum, but it is the optimum of an approximate loss function. Advantages of this second approach include the fact that the nature of the approximation is more explicit, and that the performance of the algorithm is more predictable.

To the loss function in Eq. (7). we add an L1 penalty term to obtain

$$\mathcal{L}(\mathbf{q}; \lambda, \mathbf{b}, K) = -\mathbf{q}^T\mathbf{b} + \tfrac{1}{2}\mathbf{q}^T K\mathbf{q} + \lambda \sum_j |q_j| \tag{16}$$

where q_j corresponds to the jth component of the vector $\mathbf{q}$. This use of an L1 term to promote sparse solutions is often called a lasso;[†] the approach was introduced by Tibsharini,[19] and has been further pursued by a number of authors.[20–23] Unlike the loss function in Eq. (11), which involves discrete constraints, the loss function in Eq. (16) is unconstrained. Furthermore, Eq. (16) is a convex, meaning

$$\mathcal{L}(p\mathbf{q}_1 + (1-p)\mathbf{q}_2) \leq p\mathcal{L}(\mathbf{q}_1) + (1-p)\mathcal{L}(\mathbf{q}_2) \tag{17}$$

for any $0 \leq p \leq 1$. A convex loss function has a single global minimum, and algorithms for finding this unique minimum can be very efficient.[9]

By choosing an appropriate value of λ, one can produce solutions with the desired level of sparseness. In general, using a larger λ leads to solutions with fewer nonzero components, but the relationship is not strictly monotonic. In fact, it is not always clear how best to choose an appropriate value of λ, and in practice one generally ends up optimizing Eq. (16) over a range of values of λ. This practice led Efron *et al.*[24] (following earlier work[25]) to devise the Least Angle Regression (LARS) algorithm for optimizing an L1-regularized regression problem over the whole range of different λ. The LARS approach was built for regression, but we adapt it here to the sparse linear filter problem. In Fig. 4, we show the LARS algorithm modified for optimizing a sparse matched filter. What the LARS algorithm does is very efficiently sweep over a full range of coefficient values λ, producing at each step a new optimal solution.

We remark that there are actually four LARS algorithms that we investigated. The variant shown in Fig. 4 is "least angle regression" and is the simplest to implement. LARS-lasso is a variant of LARS that actually does optimize the L1-penalized loss in Eq. (16). Where LARS is an algorithm that iteratively adds new features to the existing feature set $\mathcal{A}$, LARS-lasso enables features to be removed as well. This enables better in-sample performance, as seen in Fig. 3(f), and we see better out-of-sample performance as well (not shown).

For the two variants, LARS and LARS-lasso, there are two different ways of getting the matched filter $\mathbf{q}$. The most direct way is to use the $\mathbf{q}$ provided by the algorithm. For LARS-lasso, this is the $\mathbf{q}$ that optimizes Eq. (16). We will write LARS-**q** (or LARS-lasso-**q**) to denote this variant where $\mathbf{q}$ is provided by the LARS (or LARS-lasso) algorithm. Since this "regularized" $\mathbf{q}$ will underperform on in-sample tests, we also consider a simple alternative, which is to use the LARS algorithm to obtain $\mathcal{A}$ (this is the set of nonzero components in $\mathbf{q}$), but then to obtain $\mathbf{q}$ by directly optimizing Eq. (9): namely $\mathbf{q}_{\mathcal{A}} = K_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{b}_{\mathcal{A}}$. This variant, we write LARS-$\mathcal{A}$ (or LARS-lasso-$\mathcal{A}$).

4.3. Out of sample experimental description

While the in-sample performance of the LARS algorithms, shown in Fig. 3(f), is disappointing, compared to the sequential selection algorithms, we also looked at out-of-sample performance. As described in Section 2.4, this involves the situation in which K_{train} and K_{test} are difference covariance matrices because the algorithm is evaluated on test data that does not include the data used for training. As Fig. 5 shows, the LARS algorithms are quite competitive in their out-of-sample performance, except for very sparse models.

5. FURTHER EXPERIMENTS WITH SPARSE MATCHED FILTERS

Fig. 6(a) indicates that the "sparser" the signature vector $\mathbf{b}$, the more effective are sparse approximations to the optimal $\mathbf{q}$. What constitutes sparseness, geometrically speaking, is alignment with coordinate axes. To address the extent to which the covariance matrix exhibits alignment with these axes, we re-do the experiment using new covariance matrices, obtained by rotating the K shown in Fig. 1(b), using a transform

$$K_{\text{rotated}} = QKQ^T \tag{18}$$

where Q is a random orthogonal matrix ($QQ^T = I$). We obtain these random matrices Q from a QR decomposition[26] (keeping the "Q" part) of a matrix for which all entries are gaussian random numbers. We see, in

[†] The lasso is meant both as an acronym (Least Absolute Shrinkage and Selection Operator) and to elicit the image of variables being squeezed, as if by a loop of rope.

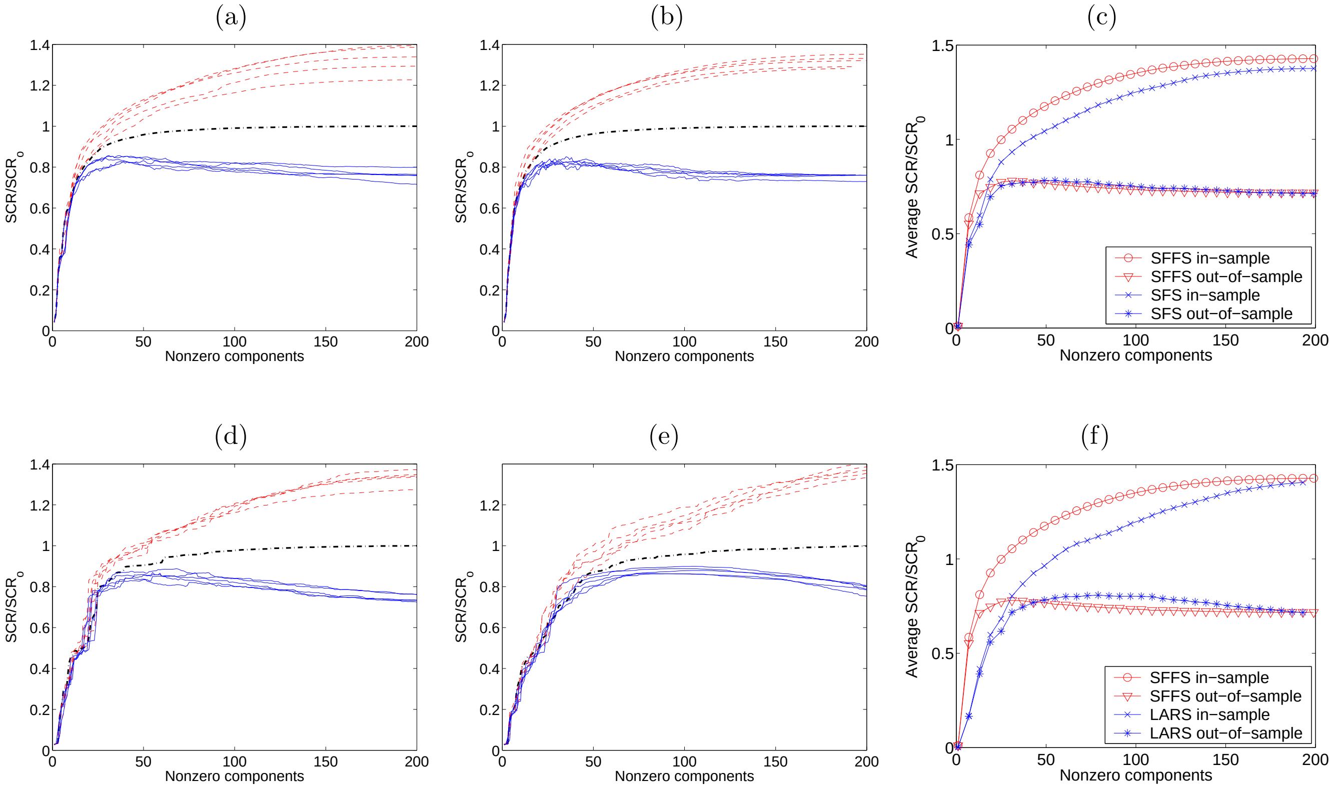

Figure 5. Comparison of in-sample and out-of-sample performance for four feature selection algorithms: **(a)** SFS, **(b)** SFFS, **(d)** LARS-lasso-$\mathcal{A}$, and **(e)** LARS-lasso-**q**. In each case, the heavy dash-dotted line is SCR obtained using the covariance entire image in Fig. 1(a). (This full-image plot corresponds to the plots shown in Fig. 3.) SCR$_o$ is defined so that these curves all saturates at 1.0. The five dashed lines above, and the five solid lines below, the full-image curves, correspond to an experiment in which covariance K_{train} is estimated from a training set of 500 pixels randomly selected from the image, and K_{test} is estimated from the remaining 15884 pixels. Sparse matched filters **q** were estimated based on these training sets, and the performance was computed both for K_{train} (upper dashed lines) and for K_{test} (lower solid lines). Also shown are averages for in-sample and out-of-sample performance: **(c)** SFFS vs SFS: although the more expensive SFFS does much better in-sample, the out-of-sample performance is nearly equal. **(f)** SFFS vs LARS: While SFFS does much better in the extremely sparse regime, LARS does better in the regime where generalization is the limiting factor.

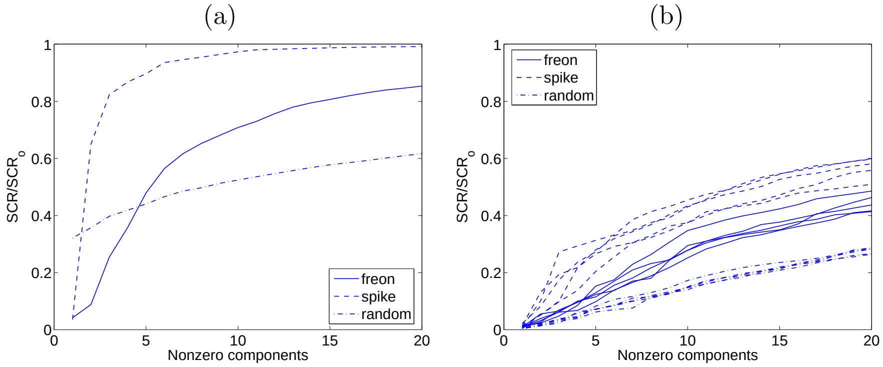

Figure 6. (a) Fraction of SCR plotted against the number of nonzero channels, for the three different signatures shown in Fig. 2. These curves are based on the stepwise forward floating selection (SFFS) scheme. **(b)** Same signatures **b** as in (a), but using covariance matrices that have been randomly rotated.

Algorithm	n=20	n=50	n=100	n=200
SFS	0.33	1.4	6.0	28
Stearns (l,r) ($l=1$, $r=2$)	0.67	3.1	15	120
SBS	104	103	100	42
SFFS	1.7	9.5	46	160
SFS-SA^x	0.68	3.7	18.8	59.4
LARS-lasso	0.19	0.27	0.49	1.5
LARS	0.14	0.20	0.38	1.4

Table 1. Runtime (in seconds) for the various sparse matched filter algorithms, implemented in Matlab, and running on a 1.7GHz Pentium M laptop, to find a model with n features. Note that sequential backward selection (SBS) takes longer to find models with fewer features. ^{x}The SFS-SA time includes the time to use SFS to obtain n features, followed by time it takes to optimize that set of n features.

Fig. 6(b) that the alignment of K to the coordinate axes is indeed a factor in how sparse a matched filter can be generated.

The diagonal normalization, described in Section 3.1, and shown in Fig. 1(c), does not affect any of the sequential selection methods, but it does affect the convex optimization algorithms, because the size of the L1 penalty will depend on the relative magnitudes of the q_i's. However, in experiments (not shown) comparing the SCR performance of the LARS algorithm with and without the diagonal normalization, we did not find a statistically significant difference.

Table 5 illustrates the relative computational efficiency of the different algorithms. SFFS is the most expensive algorithm, when it comes to computing a large number of features, but it also gives the best in-sample performance. The LARS algorithms are much more efficient, and although their in-sample performance is not as good as the heuristic sequential algorithms, it's out-of-sample performance is just as good.

6. SPARSE LINEAR DISCRIMINANT FOR CLASSIFICATION

This study is based on twelve datasets analyzed by Harvey *et al.*,[27] investigating the utility of a genetic algorithm based classifier for multispectral imagery. In Ref. 27, however, the classifier was applied to a ten-band multispectral image that had been artificially generated from AVIRIS hyperspectral imagery to mimic Multispectral Thermal Imager (MTI) data. Here, by contrast, we will use a simple Fisher discriminant as a classifier, but will apply it to the full 224-channel AVIRIS datacube.

As Fig. 7(a-d) indicates, sparse solutions can be as effective for classification as they are for weak signal detection. In these cases, we see that ninety-percent of class separability, as measured by relative SCR, can be obtained with only ten percent of the channels. Fig. 7(e) shows what bands were chosen for each of the detection problems; there is a fair bit of variability between the different datasets (and even more variability between the different classification tasks), but some patterns are evident.

7. EXTENDED SPARSE MATCHED FILTERS

With only a few hundred spectral channels, it is reasonable to find near=optimal sparse matched filters using heuristic techniques such as stepwise forward floating selection. But there are variants of the matched filter problem for which the number of potential features becomes untenably large, and more computationally efficient selection methods become a necessity. We will mention one of these variants here.

7.1. Designing multispectral imagers

For specific applications, multispectral imagery can perform nearly as well as the more flexible (and expensive) hyperspectral counterpart. In a multispectral imager, there are fewer bands, and they cover a broader wavelength range. But with careful choice of wavelength cutoffs, sensitive discrimination can be achieved. For the MTI (Multispectral Thermal Imager) sensor, for instance, these wavelength cutoffs were chosen by hand, based on careful consideration of a wide range of potential applications.[28]

Hyperspectral imagery is a great resource for investigating this problem. We can define a "band" as a contiguous set of hyperspectral channels. Each band is defined by a start channel and an end channel, and the

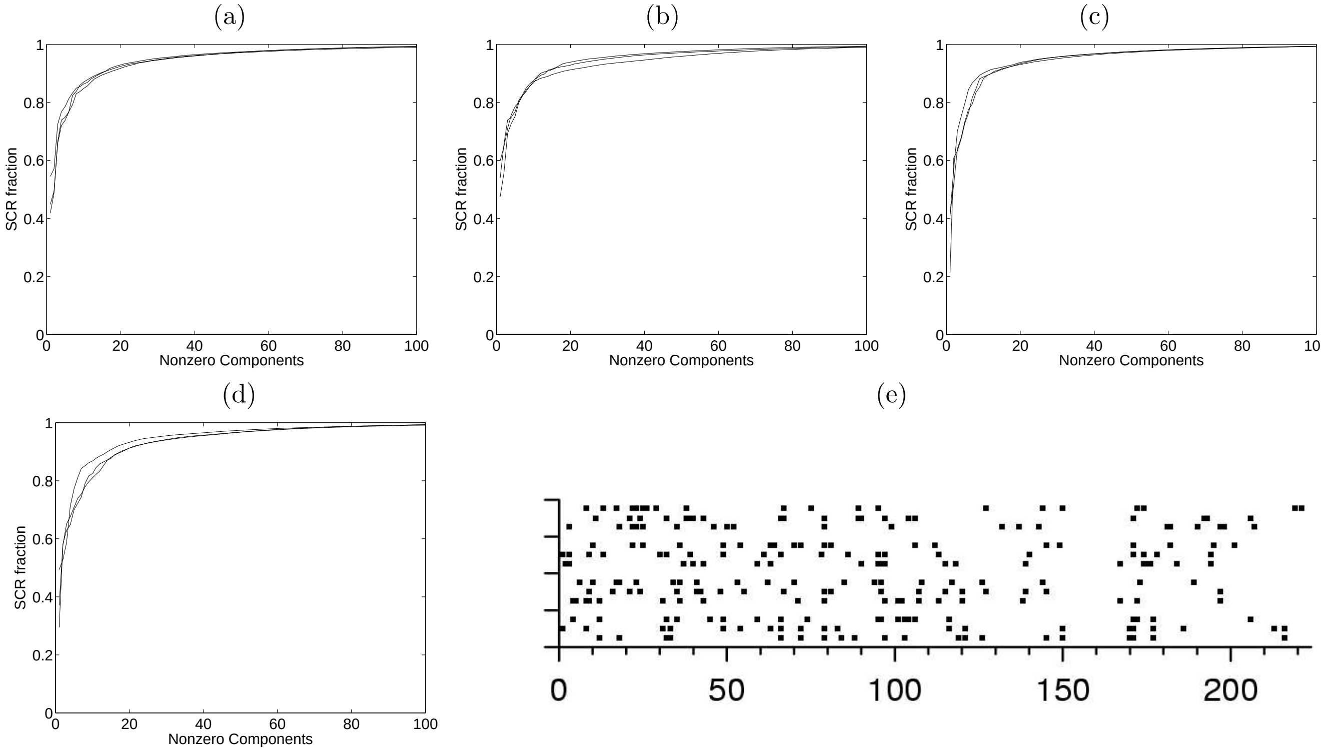

Figure 7. (a-d) Sparseness plot for Fisher discriminant, based on the twelve data sets used by Harvey *et al.*.[27] There are three different cases for each of four different target types. Here, the target types are shown in separate plots. **(a)** clouds. **(b)** golf courses. **(c)** roads. **(d)** urban areas. **(e)** For a sparse model that chooses just twenty channels, shown here are the channels that are chosen, for each of the twelve datasets in (a-d). From top to bottom, each set of three channel choices are roads, urban areas, golf courses, and clouds. It is evident that the choice of channels depends not only on the nature of the problem, but also varies considerably from instance to instance of that problem. On the other hand, there are some band ranges (such as channels 150-170) where channels are never chosen. For the clouds, for instance, which is the bottom row of three, identity or near-identity is observed in these locations: 32-33, 66, 171, 177.

signal in that band is the sum of the signals in each of the channels. Thus, with d channels in a hyperspectral image, there are $d(d+1)/2$ spectral bands that can potentially be generated.

We remark that this band selection problem (and variants thereof) has elicited considerable interest in the remote sensing community.[29–34] In a paper by Paskaleva and Hayat,[35] the performance of optimized band selection is compared directly to the performance of the MTI bands. Shen and Bassett[36] addressed this problem using an information theoretic approach. This work was a follow-on to previous work[37] which considered a smaller set of problems (anomaly detection vs anomaly and specific target ID) over a narrower range of wavelengths (0.4 to 1.5 versus 0.4 to 2.5 microns). Based on a library of material spectra, bands were selected to optimize the entropy of histograms. The optimization itself was done with a genetic algorithm.

8. DISCUSSION

There are a number of reasons to limit the number of bands in a matched filter or a discriminant model, and these reasons can be put into into two classes: convenience, and regularization. Convenient reasons include reduced computation time and storage overhead, as well as the possibility that the inclusion of fewer bands will assist with the physical interpretability of the solution. Regularization refers to schemes that reduce the complexity of the solution for the purpose of avoiding overfitting. It is well known that heroic efforts to fit an in-sample data set can lead to very poor out-of-sample performance.[38] In the words of the folk aphorism: the hurrier you go, the behinder you get. Regularization enforces a principled limitation on how hard you can try to fit data, and it can lead to improved out-of-sample performance, even as it hobbles the in-sample model.

If convenience-based reasons are the main consideration (this often occurs when there is a plentitude of training data), we find that it pays to use more aggressive (and expensive) efforts to find the best sparse models. Among the heuristic selection techniques we looked at, the sequential stepwise floating selection[14] provided the best SCR performance, and was also the most expensive. Convex optimization, in this convenience-based regime, does relatively poorly (although it is cheap), presumably because it is optimizing the wrong loss function.

But in the regularization regime, where the purpose of sparsity is to avoid overfitting, then convex optimization provides algorithms that are not only computationally much faster, their performance on out-of-sample data is comparable. In fact, using a matched filter from the LARS-lasso-**q** algorithm, which incorporates "shrinkage" as well as sparsity, we achieve better out-of-sample SCR than any of the heuristic stepwise selection algorithms. It bears remarking, however, that this out-of-sample performance is obtained by models with upwards of 100 bands – they are not, in short, very sparse.

The problem of finding *the* optimal subset of bands for a given application is NP-hard, and it seems that the only way to get improved performance is to employ increased computation. On the other hand, the problem of finding a model that works well out-of-sample is not computationally expensive. The LARS algorithm, in particular, is an efficient implementation of an adaptive L1 penalty; although it was derived for regression problems, we have shown that it can be modified for sparse filters.

In the regime of unduly many features, such as the multispectral band selection problem, online feature selection[21] may be provide an appropriate approach. Here, features are treated as an incremental resource which are evaluated and then either kept or discarded. We have investigated online feature selection in an image processing context,[39] but we speculate that it may be useful in the band selection problem as well.

REFERENCES

1. G. F. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Trans. Information Theory* **IT14**, pp. 55–63, 1968.
2. S. D. Stearns, B. E. Wilson, and J. R. Peterson, "Dimensionality reduction by optimal band selection for pixel classification of hyperspectral imagery," *Proc. SPIE* **2028**, pp. 118–127, 1993.
3. S. B. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Trans. Geoscience and Remote Sensing* **39**, pp. 1360–1367, 2001.
4. S. B. Serpico, M. D'Inca, F. Me1gani, and G. Moser, "A comparison of feature reduction techniques for classification of hyperspectral remote-sensing data," *Proc. SPIE* **4885**, pp. 347–358, 2003.
5. N. Keshava, "Angle-based band selection for material identification in hyperspectral processing," *Proc. SPIE* **5093**, pp. 440–451, 2003.
6. D. Korycinski, M. M. Crawford, and J. W. Barnes, "Adaptive feature selection for hyperspectral data analysis," *Proc. SPIE* **5238**, pp. 213–225, 2004.
7. M. Kumar, C. J. Duffy, and P. M. Reed, "Enhancing the performance of feature selection algorithms for classifying hyperspectral imagery," *Proc. International Geoscience and Remote Sensing Symposium (IGARSS)* **5**, pp. 3264–3267, 2004.
8. I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Machine Learning Research* **3**, pp. 1157–1182, 2003.
9. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, 2004.
10. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, second ed., 1990.
11. G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)," *Remote Sensing of the Environment* **44**, pp. 127–143, 1993.
12. Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), Jet Propulsion Laboratory (JPL), National Aeronautics and Space Administration (NASA) `http://aviris.jpl.nasa.gov/`.
13. S. D. Stearns, "On selecting features for pattern classifiers," in *Proc. Third Int'l Joint Conf. on Pattern Recognition*, pp. 71–75, 1976.
14. P. Pudil, J. Novovicova, , and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters* **15**, pp. 1119–1125, 1994.

15. A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **19**, pp. 153–158, 1997.
16. P. M. Narendra and K. Fukunaga, "A branch and bound algorithm for feature subset selection," *IEEE Trans. on Computers* **C-26**, pp. 917–922, 1977.
17. B. Yu and B. Yuan, "A more efficient branch and bound algorithm for feature selection," *Pattern Recognition* **26**, pp. 883–889, 1993.
18. P. Somol, P. Pudil, and J. Kittler, "Fast branch & bound algorithms for optimal feature selection," *IEEE Trans. Pattern Analysis and Machine Intelligence* **26**, pp. 900–912, 2004.
19. R. Tibsharini, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B* **58**, pp. 267–288, 1996.
20. T. Hastie, R. Tibsharini, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, Springer, New York, 2001.
21. S. Perkins, K. Lacker, and J. Theiler, "Grafting: Fast, incremental feature selection by gradient descent in function space," *J. Machine Learning Research* **3**, pp. 1333–1356, 2003.
22. S. Perkins and J. Theiler, "Online feature selection using grafting," *Proc. ICML* **20**, pp. 592–599, 2003.
23. Y. Kim and J. Kim, "Gradient LASSO for feature selection," *Proc. ICML* **21**, pp. 60–67, 2004.
24. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Stat.* **32**, pp. 407–499, 2004.
25. M. R. Osborne, B. Presnell, and B. A. Turlach, "A new approach to variable selection in least squares problems," *IMA Journal of Numerical Analysis* **20**, pp. 389–403, 2000.
26. G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 3rd ed., 1996.
27. N. R. Harvey, J. Theiler, S. P. Brumby, S. Perkins, J. J. Szymanski, J. J. Bloch, R. B. Porter, M. Galassi, and A. C. Young, "Comparison of GENIE and conventional supervised classifiers for multispectral image feature extraction," *IEEE Trans. Geoscience and Remote Sensing* **40**, pp. 393–404, 2002.
28. W. B. Clodius, P. G. Weber, C. C. Borel, and B. W. Smith, "Multi-spectral band selection for satellite-based systems," *Proc. SPIE* **3377**, pp. 11–21, 1998.
29. D. J. Wiersma and D. Landgrebe, "Analytical design of multispectral sensors," *IEEE Trans. Geoscience and Remote Sensing* **GE-18**, pp. 180–189, 1980.
30. J. C. Price, "Band selection procedure for multispectral scanners," *Applied Optics* **33**, pp. 3281–3297, 1994.
31. S. Kumar, J. Ghosh, and M. M. Crawford, "Best-bases feature extraction algorithms for classification of hyperspectral data," *IEEE Trans. Geoscience and Remote Sensing* **39**, pp. 1368–1379, 2001.
32. J. Karlholm and I. Renhorn, "Wavelength band selection method for multispectral target detection," *Applied Optics* **41**, pp. 6786–6795, 2002.
33. M. Riedmann and E. J. Milton, "Supervised band selection for optimal use of data from airborne hyperspectral sensors," *Proc. International Geoscience and Remote Sensing Symposium (IGARSS)* **3**, pp. 1770–1772, 2003.
34. S. De Backer, P. Kempeneers, W. Debruyn, and P. Scheunders, "A band selection technique for spectral classification," *IEEE Geoscience and Remote Sensing Letters* **2**, pp. 319–323, 2005.
35. B. Paskaleva and M. M. Hayat, "Optimized algorithm for spectral band selection for rock-type classification," *Proc. SPIE* **5806**, pp. 131–138, 2005.
36. S. S. Shen and E. M. Bassett, "Information theory based band selection and utility evaluation for reflective spectral systems," *Proc. SPIE* **4725**, pp. 18–29, 2002.
37. E. M. Bassett and S. S. Shen, "Information theory-based band selection for multispectral systems," *Proc. SPIE* **3118**, pp. 28–35, 1997.
38. T. G. Dietterich, "Overfitting and under-computing in machine learning," *Computing Surveys* **27**, pp. 326–327, 1995.
39. K. Glocer, D. Eads, and J. Theiler, "Online feature selection for pixel classification," *Proc. ICML* **22**, pp. 249–256, 2005.