Sparse Atomic Feature Learning via Gradient Regularization: With Applications to Finding Sparse Representations of fMRI Activity Patterns

Michael J. O'Brien*, Matthew S. Keegan*, Tom Goldstein†, Rachel Millin*,
James Benvenuto*, Kendrick Kay‡, Rajan Bhattacharyya*

* Center for Neural and Emergent Systems,
Information and System Sciences Department
HRL Laboratories LLC
Malibu, CA 90265
Emails: {mjobrien,mskeegan,rmillin,jbenvenuto,rbhattac}@hrl.com

† Computer Science Department,
University of Maryland
A.V. Williams Building
College Park, MD 20742
Email: tomgoldstein1@gmail.com

‡ Psychology Department,
Washington University in St. Louis
Campus Box 1125
One Brookings Dr
St. Louis, MO 63130-4899
Email: kendrick.kay@wustl.edu

***Abstract*—We present an algorithm, Sparse Atomic Feature Learning (SAFL), that transforms noisy labeled datasets into a sparse domain by learning atomic features of the underlying signal space via gradient minimization. The sparse signal representations are highly compressed and cleaner than the original signals. We demonstrate the effectiveness of our techniques on fMRI activity patterns. We produce low-dimensional, sparse representations which achieve over 98% compression of the original signals. The transformed signals can be used to classify left-out testing data at a higher accuracy than the initial data.**

I. Introduction

Many fields of interest are faced with the challenge of making sense of observations of noisy, repeated signals. We start with a set of N observations of a D-component signal, represented as columns of the $N \times D$ signal matrix $\mathbf{V}$. We seek a sparse decomposition of the signals into a low-dimensional linear combination of fundamental atomic features, represented by the columns of the $D \times K$ feature basis matrix $\mathbf{F}$, where $K \ll N$. The sparsely encoded signal is represented by the columns of the $K \times N$ sparse signal matrix $\mathbf{A}$. The sparsity and dimension reduction of the encoding $\mathbf{A}$ provides high signal compression. The sparse signals produced by the proposed algorithm, which we call *Sparse Atomic Feature Learning* (SAFL), are shown to be cleaner representations of the underlying class structure, resulting in a higher classification rate than the original signals as measured by a linear classifier.

The power in the proposed technique is that it leverages prior information of sparsity and repeated signal classes by using a gradient minimization approach coupled with sparsity-seeking ℓ^1 minimization, producing a variant of the fused lasso [1]. We also employ a norm-scaling technique to *model the noise*, greatly improving the results. Our transformation produces a set of basis features that can give interesting insights into the atomic features composing the original signals.

Sparse coding has previously been suggested as a mechanism for representing information in the brain [2], [3]. The encoding of visual features has been shown to employ sparse coding at different levels of the visual processing hierarchy, from V1 [4] to inferotemporal cortex [5]. Here we present the SAFL algorithm to discover underlying neural representations that takes advantage of this sparse coding. We demonstrate this capability empirically by decoding physiological measurements of neural activity from fMRI to obtain the stimulus identities giving rise to them.

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Air Force Research Laboratory (AFRL) contract number FA8650-13-C-7356. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, AFRL, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

II. Methods

In Section IV we apply SAFL, defined in Section III, to fMRI measurements of neural activity. We seek to decompose fMRI data into a set of fundamental basis features, which could be associated with semantic features of the underlying stimuli. We use *Dataset 10* from [6] because it is a large, publicly available dataset consisting of 35 different stimulus classes and 10 repetitions per class, for 350 stimulus presentations. The data consists of recorded time series for fMRI BOLD (blood-oxygen-level dependent) contrasts of the individual *voxels* that comprise the brain volume. The experimental design is described in detail in [6]. In Section II-A, we describe the preprocessing used to denoise the data and to transform the data from the temporal domain to the event-driven domain. SAFL is applied to this event-driven data, and its performance is tested using classification methods described in Section II-B.

A. Preprocessing: GLM denoising

To prepare the data, the fMRI BOLD data were divided into training and testing sets using a leave-one-run-out cross-validation design. In each fold of the cross-validation, 9 out of 10 runs (training set) were used for model optimization, and the remaining run (testing set) was used to test performance. Software provided as part of the GLMdenoise package [6] was used to estimate the hemodynamic response function (HRF) from the training data. The resulting response estimates, or *beta weights*, can be thought of as the results of a transformation from the temporal domain to the event domain. The voxel betas were stored in two matrices, $\mathbf{V}$ and $\mathbf{V}_{\text{test}}$, with the rows corresponding to voxels and the columns corresponding to stimuli.

Dimensionality was reduced by selecting the most *stable* voxels [7], meaning the voxels that demonstrated the most consistent responses across training runs. Pairwise correlations were determined between the patterns of responses (sorted by stimulus) for each pair of training runs for each voxel. The mean correlation across runs was considered the stability measure. The matrices $\mathbf{V}$ and $\mathbf{V}_{\text{train}}$ were replaced by the respective submatrices corresponding to the 500 most stable voxels from the stability analysis of the training data, $\mathbf{V}$. The results are: a 500 × 315 signal matrix $\mathbf{V}$, comprised of $N = 315$ signal responses to 35 stimuli (9 repetitions each) for $D = 500$ voxels, and a 500 × 35 testing matrix, $\mathbf{V}_{\text{test}}$, of testing signals, one for each stimulus type.

B. Classification

LIBSVM [8], a linear SVM (support vector machine) classifier, was used to produce all classification results reported in Section IV. We used LIBSVM in two different modes of classification. The first mode was to use LIBSVM's internal cross-validation tools to internally fold the data into training and testing subsets. LIBSVM then reports the percent correct over all the folds. We applied this method of classification on the training data ($\mathbf{V}$ or $\mathbf{A}$), considering the data as an aggregate signal set of nine folds. The second mode used was manually performing cross validation by explicitly training LIBSVM on a training data set ($\mathbf{V}$ or $\mathbf{A}$), and testing the trained model on the corresponding left-out testing data ($\mathbf{V}_{\text{test}}$ or $\mathbf{A}_{\text{test}}$). In this case, the predicted labels were then compared to the true labels, and the percent correct performance was reported across all ten folds.

III. Sparse atomic feature learning

A. Notation

We use to the following notation. We define $\mathbf{I}_n$ to be the $n \times n$ identity matrix. Lower case bold characters refer to vectors and upper case bold characters correspond to matrices. Matrices will consist of the component representation $\mathbf{M} = \left(m^i_j\right)$, with column vectors $\{\mathbf{m}_j\}$ and row vectors $\{\mathbf{m}^i\}$, however we may also refer to columns of $\mathbf{M}$ by $\mathbf{M}_j$, when convenient. Likewise, for an index set s, we may refer to the columns and rows of $\mathbf{M}$ indexed by s by $\mathbf{M}_s$ and $\mathbf{M}^s$, respectively. The transpose of a matrix $\mathbf{M}$ is denoted by $\mathbf{M}^T$.

We will employ an ℓ^1 norm of a matrix, $|\mathbf{M}| = \sum_{i,j} |m^i_j|$. We will also use the Frobenius norm, $||\mathbf{M}||^2_{\text{Fro}} := \text{Tr}[\mathbf{M}^T\mathbf{M}]^2 = \sum_{i,j}(m^i_j)^2$. Furthermore, we will use the Mahalanobis semi-norm, with respect to the symmetric scaling matrix, $\mathbf{S}$:

$$||\mathbf{M}||^2_{\mathbf{S}} = ||\mathbf{S}^{-1/2}\mathbf{M}||^2_{\text{Fro}} = \text{Tr}[\mathbf{M}^T\mathbf{S}^{-1}\mathbf{M}]^2.$$

We define the outer product of two vectors to be $\mathbf{u}\otimes\mathbf{v}$, where $\mathbf{u}$ is a column vector and $\mathbf{v}$ is a row vector. The gradient operator ∇_x applied to a matrix $\mathbf{M}$ is given by the matrix consisting of the column vectors $\mathbf{M}_{j+1} - \mathbf{M}_j$. Finally, $\langle x_r \rangle_{\{r\in R\}}$ is the expectation of x over the index set R.

B. Low-dimensional signal representation

We assume a finite set of stimuli, $G = \{g_1, \ldots, g_P\}$. Define the stimulus presentation vector as $\mathbf{s} = (s_1, \ldots, s_N)$ with $s_i \in G$. We assume that $\mathbf{s}$ is sorted by stimulus: for $s_i = g_k$ and $s_j = g_\ell$, $i < j$ implies $k \leq \ell$. Further, we assume that a D dimensional signal for stimulus i is measured, given by $\mathbf{v}_i = (v_1, \ldots, v_D)^T$.

Starting with a raw signal matrix $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N]$ corresponding with stimulus labels $\mathbf{s}$, the goal of this formulation is to prescribe a method of learning an atomic feature basis set, represented by the feature matrix $\mathbf{F} = [\mathbf{f}_1, \ldots, \mathbf{f}_K]$, and the corresponding coefficients. Traditionally, $\mathbf{F}$ is called a dictionary and the columns are called atoms. The representation of a signal vector $\mathbf{v}_i$ is given by the coefficient vector $\mathbf{a}_i$:

$$\mathbf{v}_i = \sum_k \mathbf{f}_k a^k_i + \eta_i = \mathbf{F}\mathbf{a}_i + \eta_i,$$

where η_i is the observation specific additive noise and $\mathbf{a}_i = (a^1_i, \ldots, a^k_i)^T$ are the coefficients that, if sparse, gate on/off the participating atomic features associated with stimulus s_i. The transformed signal $\mathbf{a}_i$ is then an encoding of the raw signal $\mathbf{v}_i$. We define $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_N]$ to be the coefficient matrix with column i corresponding to presentation $s_i \in G$. Similarly, we define the noise matrix $\mathcal{N} = [\eta_1, \ldots, \eta_N]$ and arrive at the matrix equation

$$\mathbf{V} = \mathbf{F}\mathbf{A} + \mathcal{N}.$$

The goal is to solve the minimization problem

$$(\mathbf{F}^\star, \mathbf{A}^\star) = \underset{\mathbf{F},\mathbf{A}}{\arg\min}\, ||\mathbf{V} - \mathbf{F}\mathbf{A}||^2_{\text{Fro}}. \qquad (1)$$

The solution to (1) yields an approximation $\mathbf{F}^\star$ to the atomic feature set, which serves as a basis for the observed signals, and the corresponding coefficients such that $\mathbf{F}^\star\mathbf{A}^\star \approx \mathbf{V}$. The encoding matrix $\mathbf{A}$ is a transformation of the signal matrix $\mathbf{V}$, decreasing the dimensionality of the signal from D raw values to K feature coefficients.

C. Constraining the problem

In this section we introduce $\mathbf{\Psi}$, an operator to regularize the coefficient matrix $\mathbf{A}$. We make the following two assumptions on the data, and construct $\mathbf{\Psi}$ to meet these assumptions. The first assumption is that incongruous stimulus classes, such as *rabbit* and *helicopter*, have very few features in common. To capture this characteristic in our framework, we enforce sparsity in the transformed signal matrix $\mathbf{A}$ through an ℓ^1 penalty term. This allows gating on and off of features across independent stimulus presentations. The second assumption is driven by the following. Features encoding the stimulus *rabbit* should be persistent across presentations of the *rabbit*. In our model, agreement between features for the same stimulus is given by: if $s_i = s_j = g_\ell \in G$ then $\mathbf{a}_i = \mathbf{a}_j$. To satisfy this assumption, we introduce a gradient term ∇_x, which penalizes column differences in $\mathbf{A}$. As the data is sorted in stimulus order, this amounts to penalizing whenever the adjacent signals differ.

Combining the regularization terms from our two assumptions, the resulting unconstrained optimization problem is

$$\begin{gathered}(\mathbf{F}^\star, \mathbf{A}^\star) = \operatorname*{arg\,min}_{\mathbf{F},\mathbf{A}} \frac{\mu}{2}||\mathbf{V} - \mathbf{F}\mathbf{A}||^2_{\text{Fro}} + |\mathbf{\Psi}\mathbf{A}| \\ \text{for} \quad \mathbf{\Psi} = \begin{pmatrix} \mathbf{I}_K \\ \gamma\nabla_x \end{pmatrix},\end{gathered} \tag{2}$$

where μ controls the strength of the reconstruction error penalty and the regularizer $\mathbf{\Psi}$ consists of the sparsity penalty $\mathbf{I}_K$ and gradient penalty $\gamma\nabla_x$. The scalar γ controls the ratio between gradient and sparsity penalties.

D. Modeling the noise

Inspired by [9], we explored scaling the reconstruction energy by a robust dispersion measurement matrix which approximates the inverse of the reconstruction error covariance matrix. This adjusts the error computation for approximated correlations in the data error. In scaling, we replace $||\mathbf{V} - \mathbf{F}\mathbf{A}||_{\text{Fro}}$ by $||\mathbf{V} - \mathbf{F}\mathbf{A}||_{\mathbf{S}}$, for scaling matrix $\mathbf{S}$. The generalized minimization problem becomes

$$\begin{aligned}(\mathbf{F}^\star, \mathbf{A}^\star) &= \operatorname*{arg\,min}_{\mathbf{F},\mathbf{A}} \frac{\mu}{2}||\mathbf{S}^{-1/2}(\mathbf{V} - \mathbf{F}\mathbf{A})||^2_{\text{Fro}} + |\mathbf{\Psi}\mathbf{A}| \\ &= \operatorname*{arg\,min}_{\mathbf{F},\mathbf{A}} \frac{\mu}{2}||\mathbf{V} - \mathbf{F}\mathbf{A}||^2_{\mathbf{S}} + |\mathbf{\Psi}\mathbf{A}|,\end{aligned} \tag{3}$$

where $\mathbf{S}$ can be either $\mathbf{I}$ for the non-scaled formulation, or $\mathbf{\Sigma}$ in the $\mathbf{\Sigma}$-*scaled* formulation. In the $\mathbf{\Sigma}$-scaled formulation, a *robust dispersion model* is given by $\mathbf{\Sigma}$, which is defined as follows. The *model signal* $\mathbf{m}_j = \mathbf{m}(s = g_j)$ is the median of the set of signals for stimulus g_j given by $\mathbf{m}_j = \texttt{median}(\{\mathbf{v}_i \ : \ s_i = g_j\})$. This model signal is favored because it is more robust to outlier and non-Gaussian noise, which is often observed in data of interest. Given the model signal, the robust dispersion model scaling matrix is:

$$\mathbf{\Sigma} := \left\langle \left\langle (\mathbf{v}_i - \mathbf{m}_i) \otimes (\mathbf{v}_i - \mathbf{m}_i)^T \right\rangle_{\{i \ | \ s_i = g_j\}} \right\rangle_{g_j \in G}. \tag{4}$$

The scaling matrix defined in (4) and applied to (3) effectively weights the minimization penalty applied to each signal component with respect to the expected correlated dispersion of noise in the components, as observed in the training data.

E. Algorithms for solving the system

The problem of minimizing $||\mathbf{V} - \mathbf{F}\mathbf{A}||_{\text{Fro}}$ for unknown dictionary $\mathbf{F}$ and coefficient set $\mathbf{A}$ is a well studied, non-convex problem. However, our problem is more involved due to the addition of the gradient term. Dictionary learning problems of the form $||\mathbf{V} - \mathbf{F}\mathbf{A}||_{\text{Fro}}$ are typically solved by alternating the processes for minimization with respect to the dictionary $\mathbf{D}$, and minimization with respect to the coefficients $\mathbf{A}$. We approach this problem in the same way, but additionally address the regularizer $\mathbf{\Psi}$. We call the high-level algorithm SAFL which we introduce in Algorithm 4. First we introduce the primitives of the algorithm written with respect to a general scaling matrix $\mathbf{S}$, where for our purposes $\mathbf{S}$ will either the identity in the non-scaled formulation or $\mathbf{\Sigma}$ in the scaled formulation.

1) Coefficient optimization: To solve for the coefficient matrix $\mathbf{A}$ we apply Algorithm 1, *forward-backward splitting* [10]. Forward-backward splitting was chosen to eliminate the need to compute the inverse of $\mathbf{F}^T\mathbf{F}$, which is a dense operator.

Algorithm 1 `ForwardBackwardSplitting`

```
Require: $\mathbf{V}, \mathbf{S}, \mathbf{F}_{(0)}, \mathbf{A}_{(0)}$
1: while not stopping criteria do
2:     $\widetilde{\mathbf{A}}_{(k)} = \mathbf{A}_{(k)} - \tau\mathbf{F}^T\mathbf{S}^{-1}(\mathbf{F}\mathbf{A}_{(k)} - \mathbf{V})$
3:     $\mathbf{A}_{(k+1)} = \operatorname*{arg\,min}_{\mathbf{A}} |\mathbf{\Psi}\mathbf{A}| + \frac{\mu}{2\tau}||\mathbf{A} - \widetilde{\mathbf{A}}_{(k)}||^2_{\text{Fro}}$
4: end while
5: return $\mathbf{A}_{(k)}$
```

To solve the proximal step in line 3, we use Split Bregman [11] as outlined in Algorithm 2.

Algorithm 2 `SplitBregman`

```
Require: $\mathbf{D}_{(0)} = \mathbf{B}_{(0)} = \mathbf{0}, \widetilde{\mathbf{A}}_{(k)}, \mathbf{U}_{(0)} = \widetilde{\mathbf{A}}_{(k)}$
while not stopping criteria do
    $\mathbf{U}_{(r+1)} = \operatorname*{arg\,min}_{\mathbf{U}} \frac{\mu}{2\tau}||\mathbf{U} - \widetilde{\mathbf{A}}_{(k)}||^2_{\text{Fro}} + \frac{\lambda}{2}||\mathbf{D}_{(r)} - \mathbf{\Psi}\mathbf{U} - \mathbf{B}_{(r)}||^2_{\text{Fro}}$
    $\mathbf{D}_{(r)} = \operatorname*{arg\,min}_{\mathbf{D}} |\mathbf{D}| + \frac{\lambda}{2}||\mathbf{D} - \mathbf{\Psi}\mathbf{U}_{(r+1)} - \mathbf{B}_{(r)}||^2_{\text{Fro}}$
    $\mathbf{B}_{(r+1)} = \mathbf{B}_{(r)} + (\mathbf{\Psi}\mathbf{U}_{(r+1)} - \mathbf{D}_{(r+1)})$
end while
return $\mathbf{U}_{(r)}$
```

2) Dictionary optimization: K-SVD is used to update the dictionary $\mathbf{F}$ [12], [13]. Though K-SVD is often used in conjunction with *orthogonal matching pursuit* (OMP) [14], it is a dictionary update algorithm that is independent of the coefficient optimization algorithm. The powerful aspect of K-SVD is that it preserves sparsity in the coefficient matrix. K-SVD is given in Algorithm 3.

Algorithm 3 preserves the sparsity pattern of $\mathbf{A}$ by using a mask for its zero elements corresponding to feature $\mathbf{f}_k$. An error measure is computed using the non-masked elements and while ignoring the contributions of $\mathbf{f}_k$. A best fit to compensate for the error is then determined via the minimization which can be done with SVD (justifying the name) as in [12] or using a faster alternating optimization technique as in [13].

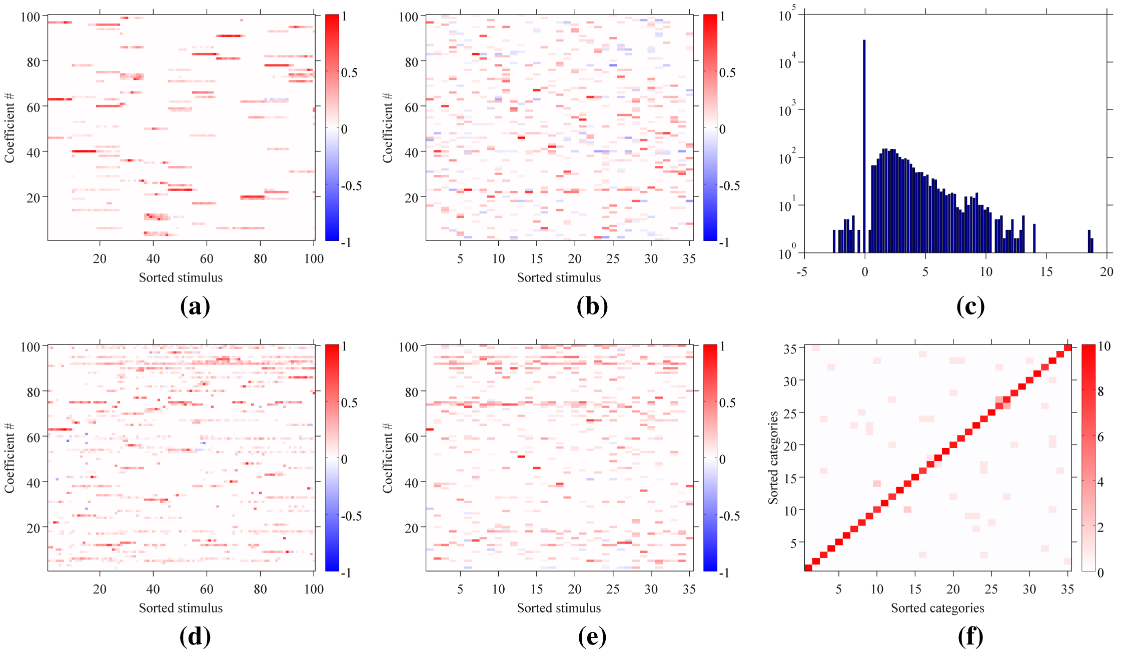


Fig. 1. Results for the sparse encoding of the first 100 signals: **(a)** $\mathbf{A}^{\mathbf{\Sigma}}$, **(b)** $\mathbf{A}^{\mathbf{\Sigma}}_{\text{test}}$, **(d)** $\mathbf{A}^{\mathbf{I}}$ and **(e)** $\mathbf{A}^{\mathbf{I}}_{\text{test}}$. The coding of the signals as the coefficient matrices are sparse from the ℓ^1 penalty term, and horizontally banded due to the gradient penalty. This banding, however, is stronger in the **(a)** $\mathbf{\Sigma}$-scaled results than the **(d)** non-scaled results. The prominence of the zero coefficients in $\mathbf{A}^{\mathbf{\Sigma}}$ are demonstrated by the log-scale histogram **(c)**, with a zero-bin of approximately 2.5 orders of magnitude greater than the other bins. **(f)** The resulting confusion matrix from the leave-one-out linear SVM cross validation testing of $\mathbf{A}^{\mathbf{\Sigma}}_{\text{test}}$, across all ten folds.

Algorithm 3 KSVD

```
Require: $\mathbf{V}, \mathbf{S}, \mathbf{F}, \mathbf{A} = (a^i_j)$
for $k = 1, \ldots, K$ do
    $idxs = \{i \ : \ a^k_i \neq 0\}$
    $\mathbf{F}_k = \mathbf{0}$
    $\mathbf{I} = \mathbf{V}_{idxs} - \mathbf{F}\mathbf{A}_{idxs}$
    $\{\mathbf{F}_k, \mathbf{A}^k_{idxs}\} = \arg\min_{\mathbf{f},\mathbf{a}} ||\mathbf{S}^{-1/2} (\mathbf{I} - \mathbf{f} \otimes \mathbf{a}) ||^2_{\text{Fro}}$
end for
return $\mathbf{F}, \mathbf{A}$
```

3) Alternate minimization: Now that we have defined our primitives for optimizing the coefficient matrix $\mathbf{A}$ and the dictionary $\mathbf{F}$, we present the high level algorithm SAFL, for solving (3) with scaling matrix $\mathbf{S} = \mathbf{I}$ or $\mathbf{S} = \mathbf{\Sigma}$. Algorithm 4 is achieved by alternating the coefficient optimization and the dictionary optimization.

Algorithm 4 SAFL

```
Require: $\mathbf{V}, \mathbf{S}$
Initialize: $\mathbf{F}, \mathbf{A}$
while not stopping criteria do
    $\mathbf{A} \leftarrow$ ForwardBackwardSplitting$(\mathbf{V}, \mathbf{S}, \mathbf{F}, \mathbf{A})$
    $\mathbf{F}, \mathbf{A} \leftarrow$ KSVD$(\mathbf{V}, \mathbf{S}, \mathbf{F}, \mathbf{A})$
end while
```

IV. Results: Application to fMRI Measurements of Neural Activity

To demonstrate the effectiveness of our approach, we apply SAFL to publicly available fMRI data [6], as described in Section II. The preprocessing of the data is described in Section II-A. Briefly, the data is transformed from temporal data to event driven neural activation values corresponding to the presented stimuli, resulting in a 500 × 315 signal matrix $\mathbf{V}$ and a 500 × 35 testing matrix $\mathbf{V}_{\text{test}}$.

In the following results we used a dictionary size of 100 with $\mu = 1$, $\lambda = 600$ and $\gamma = \frac{1}{3}$, implying a gradient penalty which is one-third of the strength of the ℓ^1 penalty. Applying Algorithm 4 using the scaling matrix defined by (4), we get the $\mathbf{\Sigma}$-scaled results $\{\mathbf{F}^{\mathbf{\Sigma}}, \mathbf{A}^{\mathbf{\Sigma}}\} \leftarrow$ SAFL$(\mathbf{V}, \mathbf{\Sigma})$. Figure 1 (a) shows $\mathbf{A}^{\mathbf{\Sigma}}$, a sparse low-dimensional representation of $\mathbf{V}$. The values of $\mathbf{A}^{\mathbf{\Sigma}}$ are also portrayed as a log-scale histogram in Figure 1 (c). Note that the bin at zero is more than 2.5 orders of magnitude larger than the other bins which, along with dimensionality reduction, achieves a compression rate of more than 98% from the original signal matrix $\mathbf{V}$. For comparison, we also generated the non-scaled results $\{\mathbf{F}^{\mathbf{I}}, \mathbf{A}^{\mathbf{I}}\} \leftarrow$ SAFL$(\mathbf{V}, \mathbf{I}_D)$, with the resulting coefficients shown in Figure 1 (d). The non-scaled result does not demonstrate banding as strongly as the $\mathbf{\Sigma}$-scaled result due to the lack of scaling by the voxel-specific data dispersion model.

In Figure 2, the $\mathbf{FA}$ activity reconstruction results are shown for both the $\mathbf{\Sigma}$-scaled (a), and non-scaled (c) formulations, with comparisons drawn from the original signal matrix (b). It is interesting that the reconstruction is more faithful in the non-scaled formulation, but the banding is much stronger in the $\mathbf{\Sigma}$-scaled model. In both models, the banding is a result of forced similarity in adjacent (same-stimulus) signals. However, the $\mathbf{\Sigma}$-scaled results are a product of the voxel specific scaling which compensates penalties for expected errors, allowing for more freedom in the model fitting. As can be seen in (a) the

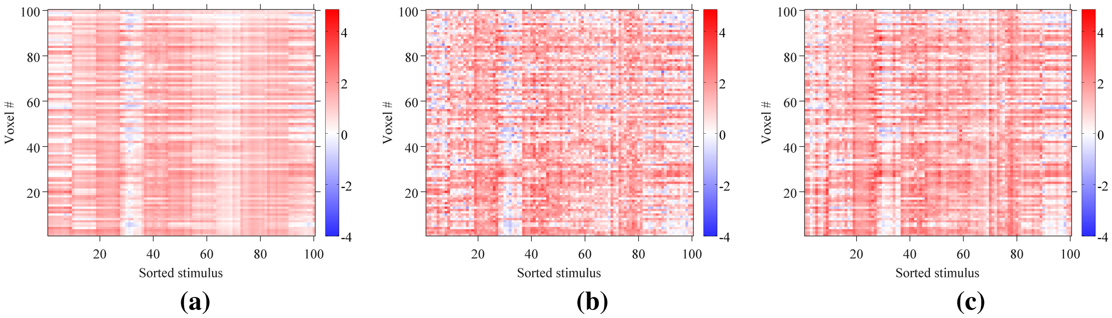


Fig. 2. Reconstruction results. The column indices correspond to the first 100 signal representations and the rows correspond to the first 100 voxels. **(a)** Reconstruction of the neural signals by $\mathbf{F}^{\mathbf{\Sigma}}\mathbf{A}^{\mathbf{\Sigma}}$, **(b)** the recorded neural signals $\mathbf{V}$, **(c)** reconstruction of the neural signals by $\mathbf{F}^{\mathbf{I}}\mathbf{A}^{\mathbf{I}}$. The non-scaled solution **(c)** can very accurately reconstruct the original neural signals, including the non-stimulus specific activity. On the other hand, the $\mathbf{\Sigma}$-scaled solution **(a)** demonstrates stronger blocks of homogeneous signal that would be expected for stimulus-sorted data. This could be interpreted as a cleaner signal with non-stimulus specific activity minimized.

scaling removes non-stimulus specific activity and provides for a cleaner signal than the non-scaled results of (c). Which is also supported by the classification results below.

Using LIBSVM's internal leave-one-out cross validation linear classifier, we get a correct classification rate of 92% for $\mathbf{V}$, 94% for $\mathbf{A}^{\mathbf{I}}$ and 100% for $\mathbf{A}^{\mathbf{\Sigma}}$. These classification results are only valid when SAFL is used purely as a sparse data compressing transformation. These classification results demonstrate that $\mathbf{\Sigma}$-scaled SAFL not only preserves distinguishable signals, but the transformation also decreases non-stimulus specific activity that is pervasive in the original patterns.

SAFL, however, is a supervised learning algorithm since it relies on sorted data. Thus, to perform cross-validation tests for data with truly unknown labels it is important to exclude testing data from the original transformation. In this situation a dictionary is learned from the training data as before but the sparse testing signal $\mathbf{A}_{\text{test}}$ is computed separately from the left-out testing data $\mathbf{V}_{\text{test}}$. We proceed by generating the testing data by $\mathbf{A}^{\mathbf{\Sigma}}_{\text{train}} \leftarrow \texttt{ForwardBackwardSplitting}(\mathbf{V}_{\text{train}}, \mathbf{\Sigma}, \mathbf{F}^{\mathbf{\Sigma}})$ and $\mathbf{A}^{\mathbf{I}}_{\text{train}} \leftarrow \texttt{ForwardBackwardSplitting}(\mathbf{V}_{\text{train}}, \mathbf{I}_D, \mathbf{F}^{\mathbf{I}})$, visualized in Figure 1 (b) and (e) respectively. The testing data is processed without class labels so ordering the data to leverage the gradient is not possible. Though other options are plausible for coefficient optimization on the testing data, we use the same forward-backward splitting (Algorithm 1) with $\gamma = 0$ to remove the gradient penalty. All other parameter are as before.

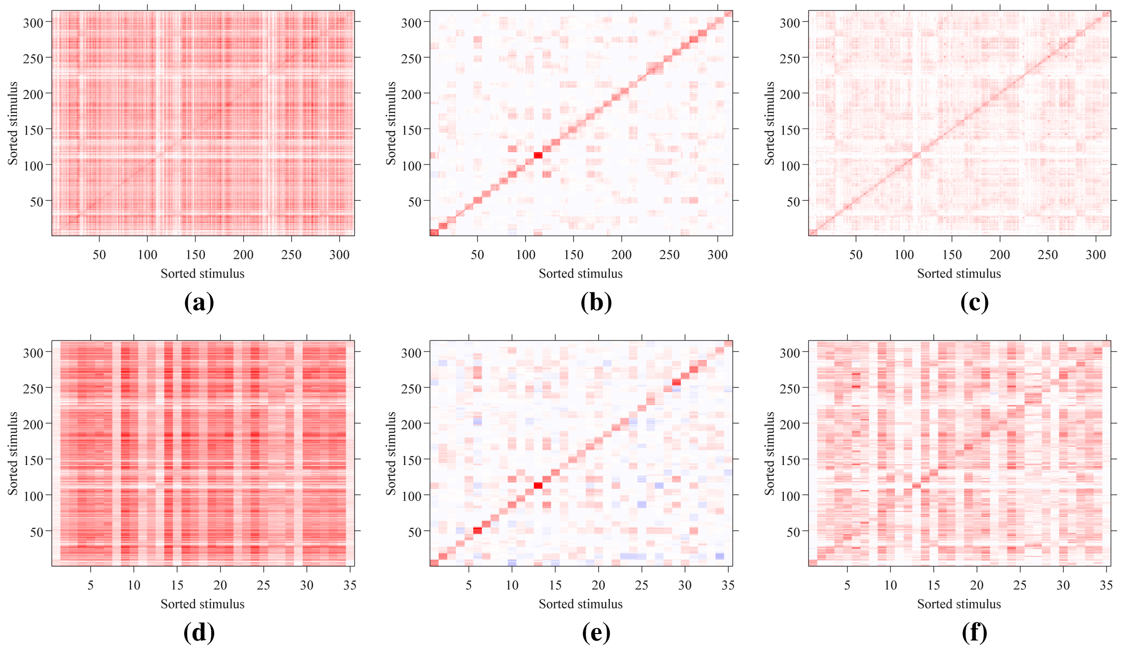
With the testing data separated in this way we use the separate `svmtrain` and `svmpredict` methods of LIBSVM on the training and testing data respectively. After repeating the procedure for all 10 folds the classification success rates were 89% for $\mathbf{V}_{\text{test}}$, 62% for $\mathbf{A}^{\mathbf{I}}_{\text{test}}$ and 93% for $\mathbf{A}^{\mathbf{\Sigma}}_{\text{test}}$. The confusion matrix for $\mathbf{A}^{\mathbf{\Sigma}}_{\text{test}}$ is shown in Figure 1 (f). The decrease in classification rate for the raw signals (from 92% to 89%) is due to the preprocessing of the data, which is itself supervised in the choice for the design matrix and learning the HRF [6]. Preprocessing $\mathbf{V}_{\text{test}}$ cannot leverage the data labels as is done in preprocessing $\mathbf{V}$ which results in more noise in the final raw signal set $\mathbf{V}_{\text{test}}$.

The superiority of the $\mathbf{\Sigma}$-scaled over the non-scaled results can be explained by the heuristic depicted in Figure 3. This figure shows signal similarity as measured by the dot product. The top three figures indicate the self-similarity of the signal matrices $\mathbf{V}$, $\mathbf{A}^{\mathbf{\Sigma}}$ and $\mathbf{A}^{\mathbf{I}}$, in order. Likewise, the bottom three figures indicate the similarity between the training signal matrix and the testing signal matrix, in the same order. In the non-scaled results (third column) it appears to be more difficult to distinguish between signals, due to generally higher similarity, than in the second column derived from the $\mathbf{\Sigma}$-scaled formulation. This agrees with the SVM classification from above. Similarity matrices can only be used as a heuristic, however, since the first column corresponds to the raw values which classifies better than the resulting non-scaled signal under SVM, yet appear more difficult to distinguish.

To test robustness of SAFL, we varied the dictionary size from 50 features to 250 features, γ from 200 to 1800, and λ from $\frac{1}{8}$ to $\frac{5}{4}$. The average classification rate on the testing data through all these experiments was 88%, with a minimum of 79% and a maximum of 93%. Though the study was not a full exploration of the parameter space, the typically high results and variety of parameters used indicate finding effective parameters is not difficult. Moreover, to test for initialization sensitivity, SAFL was repeated ten times with different data initializations resulting in no more than a 2% change in final classification rate.

V. Discussion & Conclusion

Given a set of noisy signals, $\mathbf{V}$, we demonstrated a machine learning framework that trains a dictionary and produces a low-dimensional, sparse signal representation, $\mathbf{F}$ and $\mathbf{A}$ respectively, such that $\mathbf{V} \approx \mathbf{FA}$. Our framework provides powerful compression of the original training data as well as unlabeled testing data $\mathbf{V}_{\text{test}}$. Beyond signal compression, the resulting sparse signals were also shown to produce better classification results than the raw values when evaluated by a linear SVM classifier, demonstrating the clean signal extraction capabilities of our framework which minimizes non-stimulus specific activity. The atomic features learned by SAFL provide further evidence that sparse coding is used for neural representations of visual objects and scenes. In future work, we plan to examine how this decodable sparse code shares atomic features between related stimuli, and individual differences for the atomic features and sparse coding.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Fig. 3. Signal similarity analysis from pairwise inner-products of the signals providing a visual heuristic of how well signals can be differentiated. In each image $\mathbf{X}^T\mathbf{Y}$ is shown for signal matrices $\mathbf{X}$ and $\mathbf{Y}$. The component at (i, j) is a measure of similarity between signals $\mathbf{X}_i$ and $\mathbf{Y}_j$. The top row is the self-similarity matrices for: **(a)** $\mathbf{V}$, **(b)** $\mathbf{A}^{\boldsymbol{\Sigma}}$, **(c)** $\mathbf{A}^{\mathbf{I}}$. The bottom row is the train-to-test similarity matrices for: **(d)** $\mathbf{V}$ and $\mathbf{V}_{\text{test}}$, **(e)** $\mathbf{A}^{\boldsymbol{\Sigma}}$ and $\mathbf{A}^{\boldsymbol{\Sigma}}_{\text{test}}$, **(f)** $\mathbf{A}^{\mathbf{I}}$ and $\mathbf{A}^{\mathbf{I}}_{\text{test}}$. We would expect a block diagonal similarity matrix for highly distinguishable signals as like-stimuli should produce similar signals and differing stimuli should produce different signals. Note that though the original signals $\mathbf{V}$, portrayed in **(a)** and **(d)**, do not give a block-diagonal dominant matrix the data still classifies well with SVM so similarity matrices can only be used as a heuristic. However the transformed signals for the non-scaled model, **(c)** and **(f)**, do not classify nearly as well while those from the $\boldsymbol{\Sigma}$-scaled transformed, **(b)** and **(e)**, outperforms the original signal, as one might expect from its strong block-diagonal similarity matrices.

Though sparsity with gradient minimization in one dimension has been proposed before in the fused lasso [1], SAFL differs in that a two dimensional data structure is used with gradient minimization in only one direction, increasing the difficulty of the problem. Split Bregman has also been proposed for solving the system [15], however we found that in two dimensions the required operator inversion was too costly. SAFL first uses forward-backward splitting to decouple the problem, resulting in an easy gradient descent step which eliminates the need for a difficult inversion in the Split Bregman routine.

References

[1] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, vol. 67, no. 1, pp. 91–108, 2005.

[2] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[3] ——, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[4] W. E. Vinje and J. L. Gallant, "Sparse coding and decorrelation in primary visual cortex during natural vision," *Science*, vol. 287, no. 5456, pp. 1273–1276, 2000.

[5] M. P. Young and S. Yamane, "Sparse population coding of faces in the inferotemporal cortex," *Science*, vol. 256, no. 5061, pp. 1327–1331, 1992.

[6] K. N. Kay, A. Rokem, J. Winawer, R. F. Dougherty, and B. A. Wandell, "GLMdenoise: a fast, automated technique for denoising task-based fMRI data," *Frontiers in Neuroscience*, vol. 7, no. December, pp. 1–15, Jan. 2013.

[7] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just, "Predicting human brain activity associated with the meanings of nouns," *Science*, vol. 320, no. 5880, pp. 1191–1195, 2008.

[8] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1—-27:27, 2011.

[9] S. Nishimoto, A. T. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. L. Gallant, "Reconstructing Visual Experiences from Brain Activity Evoked by Natural Movies," *Current Biology*, vol. 21, no. 19, pp. 1641–1646, Oct. 2011.

[10] P. Combettes and J. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer Optimization and Its Applications, 2011, pp. 185–212.

[11] T. Goldstein and S. Osher, "The Split Bregman Method for L1-Regularized Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, Jan. 2009.

[12] M. Aharon, M. Elad, and A. Bruckstein, "K -SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[13] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit," CS Technion, Tech. Rep., 2008.

[14] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, vol. 1, 1993.

[15] G.-B. Ye and X. Xie, "Split Bregman method for large scale fused Lasso," *Computational Statistics & Data Analysis*, vol. 55, no. 4, pp. 1552–1569, Jun. 2011.