SPARSE SENSOR PLACEMENT OPTIMIZATION FOR CLASSIFICATION

B. W. BRUNTON, S. L. BRUNTON, J. L. PROCTOR, AND J. N. KUTZ

Abstract. Choosing a limited set of sensor locations to characterize or classify a high-dimensional system is an important challenge in engineering design. Traditionally, optimizing the sensor locations involves a brute-force, combinatorial search, which is NP-hard and is computationally intractable for even moderately large problems. Using recent advances in sparsity-promoting techniques, we present an algorithm to solve this sparse sensor placement optimization for classification (SSPOC) that exploits low-dimensional structure exhibited by many high-dimensional systems. Our approach is inspired by compressed sensing, a framework that reconstructs data from few measurements. If only classification is required, reconstruction can be circumvented and the measurements needed are orders-of-magnitude fewer still. Our algorithm solves an \( \ell_1 \) minimization to find the fewest nonzero entries of the full measurement vector that exactly reconstruct the discriminant vector in feature space; these entries represent sensor locations that best inform the decision task. We demonstrate the SSPOC algorithm on five classification tasks, using datasets from a diverse set of examples, including physical dynamical systems, image recognition, and microarray cancer identification. Once training identifies sensor locations, data taken at these locations forms a low-dimensional measurement space, and we perform computationally efficient classification with accuracy approaching that of classification using full-state data. The algorithm also works when trained on heavily subsampled data, eliminating the need for unrealistic full-state training data.

Key words. sensor placement optimization, classification, compressed sensing, \( \ell_1 \) minimization, enhanced sparsity, feature selection, machine learning

AMS subject classifications. 49N30, 94A13, 94A20

DOI. 10.1137/15M1036713

1. Introduction. Sensors play a vital role in characterizing high-dimensional, nonlinear systems in science and engineering, providing measurements to inform one or more of the following tasks: (1) classification or detection, (2) state estimation or reconstruction, and (3) future state prediction. The type of sensor determines measurement quality, temporal resolution, and latency, while sensor locations can be chosen to gather specific information about relevant spatial features in the system. It is often necessary to make use of limited numbers of sensors, either because individual measurements are expensive to acquire, or computational constraints make it impossible to process full-state measurements in real-time. The cost associated with...
placing sensors is a limiting factor for sampling in many domains, including ocean and atmospheric science, epidemiology and disease control, and neuroscience. Similarly, in many engineering tasks, it is increasingly important to make fast control decisions based on efficient low-order models, with a few key sensors placed strategically to gather information and manage uncertainty.

Designing and optimizing a limited set of sensor locations is an important unsolved challenge. In the most straightforward approach, optimizing the placement of sensors for a particular task with a well-defined cost function amounts to a combinatorial search. This search is exhaustive and NP-hard, and thus computationally intractable for all but the simplest examples [2, 25]. Because of the brute-force nature of this optimization, prospects of scaling to larger systems do not improve with exponentially increasing computational resources. To tackle this challenge, we are inspired by advances in the theory of compressed sensing [11, 19, 2], which make it possible to solve this type of combinatorial optimization using algorithms that scale well with size. Importantly, compressed sensing takes advantage of a signal’s compressibility. More generally, most high-dimensional dynamical systems exhibit low-dimensional patterns and coherent structures that facilitate these sparse sensing strategies [52, 8, 46].

Compressed sensing is able to reconstruct a signal using surprisingly few measurements, but assigning a signal to one of a few categories may be accomplished with orders-of-magnitude fewer measurements (see, e.g., [61]). We refer to this reduction in the number of measurements required for classification over reconstruction as enhanced sparsity. For example, typical natural images with $n$ pixels can be recovered by compressed sensing with significant savings, usually from $n/10$ to $n/3$ measurements [50]. In contrast, as we will see, classification of a natural image may require only tens of measurements. More generally in engineering applications, machine learning and classification are becoming more widely used for modeling and control [1, 27, 33, 7].

Classification is often performed on a tailored low-dimensional basis. This basis is frequently formed by hierarchical features of the data that may be ordered by energy or variance. One of the most ubiquitous methods in dimensionality reduction is principal component analysis (PCA) [44, 20], which may be computed by a singular value decomposition (SVD) to yield an ordered orthonormal basis. High-variance PCA modes can account for common features shared across categories, which are not useful for discrimination. Linear discriminant analysis (LDA) [22, 49, 4] is commonly applied in conjunction with PCA to find the relevant features that are useful for discrimination. Indeed, PCA-LDA is a standard approach in machine learning [20], and it is still broadly used today because of its simplicity and effectiveness for a wide range of problems. For this reason, the methods developed in this paper are demonstrated on PCA-LDA classification, although they are readily generalizable to other feature extraction and classification methods.

In this work, we address the problem of optimal sensor placement for categorical tasks. In particular, we leverage the sparsity-promoting $\ell_1$ minimization to solve an underdetermined system of equations for the sparsest measurement vector that reconstructs the low-dimensional discriminant space in feature space; sparse sensors are placed at nonzero entries of the sparse measurement vector.

1.1. Related work. There has been significant work combining ideas from compressed sensing with classification to achieve decisions based on few measurements. For example, compressive classification has been explored with using a small number of random measurements to achieve classification without reconstruction [17]. The
sparse representation algorithm has been applied to biometric recognition problems and is surprisingly robust [61, 45]. Sparse approximation applied to semantic hierarchies [41] has been shown to be efficient in categorizing between large numbers of classes [35]. Ideas from compressed sensing have also been applied to dimensionality reduction tools to develop a theory of sketched SVD based on randomly projected data [23, 47, 29]. Methods to promote sparsity and parsimonious representations of high-dimensional data from few measurements include the development of sparse PCA (SPCA) [64] and, in a related line of work in feature selection for classification, penalized and sparse LDA [60, 16].

Many applications that require high-performance computing to model dynamical processes in physical systems rely on reduced-order models (ROMs) [48, 3] and sensor refinement to streamline computations. As a precursor to modern sparse sensing theory, Everson and Sirovich [21] introduced a method for computing the proper orthogonal decomposition (POD) [31] using incomplete or gappy measurement data. In this gappy POD technique, a small number of measurements are used to evaluate nonlinear terms in the governing equations and construct POD-Galerkin approximations of complex systems [37]. More recently, the low-rank POD approximations were combined with sparse representation to enable ROM computations with very limited sensors [5, 8]. Significant effort has been applied to develop principled, rather than random, sensor placement for ROMs [59, 63, 15, 51]. Although none of the above methods solve the typically NP-hard sensor placement optimization problem, they have been demonstrated to be quite effective in many applications.

1.2. Contributions and perspectives of this work. In this work, we develop an algorithm to optimally select, within a large set of possible measurement locations, a smaller subset of key locations that best inform a classification task. It is a novel framework to harness enhanced sparsity in application domains where it is economical to measure at as few sensor locations as possible. We demonstrate that classifiers using very few learned sensors perform comparably with classifiers using the full-state data. Further, the algorithm includes a parameter to tune the tradeoff between number of sensors and accuracy.

Datasets may be vast, and data acquisition can be limited by bandwidth on data streams. Therefore, we also develop an intermediate technique, related to the sketched SVD [23, 28, 29], to start instead with a subsample of the original data. We demonstrate that starting with 10% of the original data, we are still able to find nearly optimal sparse sensor locations.

Our framework for sparse classification has two characteristics that distinguish it significantly from previous work: (1) instead of random projection measurements, physically realizable point sensor locations are selected specifically, and (2) \( \ell_1 \) minimization is applied once in sensor learning, and classification of new samples involves inner products on a very small measurement vector. A schematic representation of our procedure is found in Figure 1, with details following in the text. Equations (3.1) and (3.3) comprise our primary theoretical contributions. Figure 3 summarizes the sensor locations produced by the algorithm for the five examples described in section 4.

This work takes an applied perspective of probing complex systems with underlying low-dimensional structure, with the explicit goal of forming some categorical decision about the state of the system. The optimally sparse spatial sensors framework is particularly well suited for engineering applications, as an upfront investment in the learning phase allows remarkably efficient performance for all subsequent queries. In addition, many engineering systems are sparse in a spatial Fourier basis, facilitating
effective incoherent sampling using a few physically realizable point measurements. This is a refreshing alternative to random projection measurements, which may be impractical in real-world applications where many individual point measurements are prohibitively expensive.

We speculate that the principle of enhanced sparsity may also have relevance for biological organisms, which often need to make decisions based on limited sensory information [6]. Specifically, organisms interact with high-dimensional physical systems in nature but must rely on information gathered through a handful of sensory organs. Our sparse sensor algorithm provides one approach to answer the question, Given a fixed budget of sensors, where should they be placed to optimally inform a decision?

1.3. Organization of the paper. Section 2 provides an overview of well-known techniques on which we build our contributions, and establishes the notation used for the remainder of the paper. We summarize compressed sensing, random projections for matrix decomposition, and the PCA-LDA method for categorization. Section 3 describes our algorithm for determining optimally sparse sensor locations by using ℓ1 minimization and a dictionary of learned features.

We applied sparse sensor placement optimization for classification (SSPOC) to five examples based on real-world data from different application domains. We chose these examples to demonstrate the potential application of SSPOC to both dynamic and static data, as well as to emphasize the interpretation of a “sensor” as a generalized measurement. Two of the examples seek to classify dynamic regimes from numerical solutions to physical dynamical systems; they are the cubic-quintic Ginzburg–Landau equation (CQGLE) and pressure fields of fluid flow around a cylinder associated with different Reynolds numbers. Two further examples are image recognition tasks—one of cat/dog classification and one of human face recognition. The last example is a microarray dataset from cancer tissue, where the task is to classify the cancer type. The results of these experiments are presented in section 4, and the methods are discussed more generally in section 5.

2. Background. The main contributions of this work are (1) to design optimal sensor locations optimized to inform classification and (2) to design sensors from substantially subsampled data. To put our contributions in context, we review well-known results from compressed sensing in section 2.1.

We also make use of two well-established methods for systematically producing low-rank representations of a high-dimensional dataset, PCA and LDA. In this paper, we demonstrate our algorithm by applying PCA in combination with LDA for classification. In general, the method can be implemented with many other dimensionality reduction and discrimination algorithms, as drawn schematically in the top of Figure 1. Section 2.3 reviews PCA-LDA and establishes the notation used to describe our methods in section 3.

2.1. Compressed sensing. Compressed sensing theory states that if the information of a signal \( x \in \mathbb{R}^n \) is \( k \)-sparse in a transformed basis \( \Psi \) (all but \( k \) coefficients are zero) and \( k \ll n \), it is possible to reconstruct the signal from very limited measurements \( O(k \log(n/k)) \) [9, 11, 14, 10]. This technique has widespread applications, including the fields of medical imaging [40], neuroscience [24], and engineering [30].

The standard framing of the compressed sensing problem is as follows. Let us suppose the measurement vector \( \tilde{x} \in \mathbb{R}^p \) is related to the desired signal \( x \in \mathbb{R}^n \) by \( \tilde{\Phi} \), a known \( p \times n \) measurement matrix: \( \tilde{x} = \tilde{\Phi}x \). Compressed sensing seeks to recover \( x \) from \( \tilde{x} \) in underdetermined cases where \( p \ll n \).
The signal $x$ has coefficients $a$ in basis $\Psi$, so that

$$\tilde{x} = \tilde{\Phi}x = \tilde{\Phi}\Psi a = \Theta a.$$  \hfill(2.1)

If $x$ is sufficiently sparse in $\Psi$ and the matrix $\Theta$ obeys the restricted isometry principle (RIP) [12], the search for $a$ (and thus the reconstruction of $x$) is possible.

We would like to solve for the sparsest $a$,

$$a = \arg\min_{a'} \|a'\|_0, \quad \text{subject to } \tilde{x} = \Theta a',$$

but this involves an intractable combinatorial search. It has been shown that under certain conditions, (2.2) may be solved by minimizing the $\ell_1$ norm [11, 18],

$$a = \arg\min_{a'} \|a'\|_1, \quad \text{subject to } \tilde{x} = \Theta a'.$$

$$\hfill(2.3)$$
Relaxation to a convex $\ell_1$ minimization bypasses the combinatorially difficult problem. Solutions to (2.3) may be found through convex optimization routines, or by greedy algorithms such as orthogonal matching pursuit [57, 56].

An important aspect of compressed sensing is the choice of the measurement matrix $\tilde{\Phi}$. To achieve exact reconstruction, $\tilde{\Phi}$ must be incoherent with respect to $\Psi$ (thereby satisfying the RIP). Many authors describe the properties of random matrix ensembles that fulfill these conditions with high probability: they include Gaussian, Bernoulli, and random partial Fourier matrices [9, 13, 19]. In addition, sparse random matrices, where many of the entries of the measurement vectors are zero, also allow reconstruction [28]. Theoretical limits of how sparse random projections may be and still support signal recovery have been explored [39, 58].

2.2. Random projections and matrix decomposition. In addition to the random measurement framework for reconstruction, we are concerned with the related question of the spectral properties of a data matrix under random projection. Consider a data matrix $X = [x_1 \ x_2 \ \ldots \ x_m] \in \mathbb{R}^{n \times m}$. Let

$$\tilde{X} = \tilde{\Phi}X,$$

where $\tilde{\Phi} \in \mathbb{R}^{p \times n}$ and $p \ll n$, as above. Spectral properties of $X$ and $\tilde{X}$ are approximately equal when the random measurement matrix $\tilde{\Phi}$ satisfies the distributed Johnson–Lindenstrauss (JL) property [32, 29]. This property regards the preservation of relative distances between points after random projection to a new space. Since the classification task depends on a geometric separation of data points between classes, the JL property is integral to the success of this model reduction step (dashed arrow between the full and subsampled PCA feature spaces in Figure 1).

2.3. PCA, LDA, and classification. Here we describe the two main dimensionality reduction techniques used in this paper to demonstrate the sensor location refinement algorithm in section 3. Consider a set of observations $x_i, i = 1, \ldots, m$, where $x_i \in \mathbb{R}^n$; also suppose that $m \ll n$. Let us construct a data matrix $X$ with columns $x_i$, as represented schematically in Figure 1; we assume that rows of $X$ have been mean subtracted. The PCA consists of performing an SVD defined by the following modal decomposition:

$$(2.4) \quad X = \Psi \Sigma V^*.$$ 

The columns of $\Psi$ are the left singular vectors of $X$: they span the columns of $X$ and are often referred to as the principal components or features of the dataset. The matrix $\Sigma$ is diagonal; entries $\sigma_{ii}$ are the singular values of $X$. There are only $m$ nonzero singular values, and we may write $\Sigma = [\Sigma_{m \times m} \ \ 0]^T$.

The columns of the matrix $\Psi$ are eigenvectors of $XX^T$:

$$XX^T \Psi = \Psi \Lambda, \quad \Lambda = [\Sigma_{m \times m}^2 \ 0 \ 0].$$

However, this is an extremely expensive eigendecomposition, since the dimension of $XX^T$ is $n \times n$. In implementation, it is more practical to use the method of snapshots [53], whereby we solve the $m \times m$ eigendecomposition for $V$:

$$X^TXV = V \Sigma_{m \times m}^2.$$
We may then obtain the $m$ nonzero PCA modes by taking a linear combination of the columns of $X$:

$$
\Psi_m = X \Sigma_{m \times m}^{-1}.
$$

The SVD provides a systematic way to transform data into a lower-dimensional representation. For datasets that are inherently low-rank, the singular values $\Sigma$ of (2.4) contain many entries that are exactly zero. With most realistic applications and datasets, though, the singular values often exhibit a power-law decrease, where the low-rank representation of the data is approximate. A power-law decrease of singular values provides an opportunity for a heuristic, and at least quantitatively informed, decision about the number of information-rich dimensions to retain for the reduced-order subspace. Taking the $r$ columns of $\Psi$ as basis vectors corresponding to the $r$ largest singular values, a truncated basis $\Psi_r$ can be formed. We may use this basis to project data from the full measurement space into the reduced $r$-dimensional PCA space (known as feature space):

$$
\Psi_r^T : \mathbb{R}^n \rightarrow \mathbb{R}^r,
$$

$$
x \mapsto a.
$$

Thus, the principal components $\Psi_r$ minimize the $\ell_2$ projection error, $\|x - \Psi_r \Psi_r^T x\|_2$.

For categorical decisions, we apply the well-known classifier LDA in $r$-dimensional feature space. Despite the plethora of classifier techniques, we chose LDA for its simplicity and well-established behavior. We note once again that PCA and LDA are serving only to illustrate the concept proposed in Figure 1.

We start by noting that each observation $x_i$ (and thus each $a_i = \Psi_r^T x_i$) belongs to one of $c$ distinct categories $C_j$, where $j = 1, \ldots, c$. LDA attempts to find a set of directions in feature space $w \in \mathbb{R}^{r \times c-1}$, where the between-class variance is maximized and the within-class variance is minimized. Specifically,

$$
w = \arg \max_w \frac{w^T S_B w'}{w'^T S_W w'}.
$$

Here $S_W$ is the within-class scatter matrix,

$$
S_W = \sum_{j=1}^{c} \sum_{i \in C_j} (a_i - \mu_j)(a_i - \mu_j)^T,
$$

where $\mu_j$ is the mean of class $j$. $S_B$ is the between-class scatter matrix,

$$
S_B = \sum_{j=1}^{c} N_j (\mu_j - \mu)(\mu_j - \mu)^T,
$$

where $N_j$ is the number of observations in class $j$ and $\mu$ is the mean of all observations in $X = \Psi_f^T X$.

It follows that $w$ are eigenvectors corresponding to the nonzero eigenvalues of $S_W^{-1} S_B$. It should be noted that the number of nontrivial LDA directions must be less than or equal to the number of categories minus one ($c - 1$), and that at least $r + c$ samples are needed to guarantee $S_W$ is not singular [4].

In schematic form, the top “full measurements” row of Figure 1 illustrates the PCA-LDA process for the classification task. In the case of $c = 2$ categories, the LDA
projection is \( w^T : \mathbb{R}^r \rightarrow \mathbb{R} \), and we may apply a threshold value that separates the two categories. Assuming the two categories have equal covariances, we pick the threshold to be the midpoint between the means of the two classes, \( w^T \mu_A \) and \( w^T \mu_B \).

For decisions between \( c \geq 2 \) categories, LDA produces a projection \( w^T : \mathbb{R}^r \rightarrow \mathbb{R}^{c-1} \). We use a nearest centroid (NC) method for classification, in which a new measurement \( x_i \) is assigned to category \( j \), for which the distance between \( w^T x_i \) and \( w^T \mu_j \) is minimal. Many other classifier choices are possible in the reduced \( \mathbb{R}^{c-1} \) space, including nearest neighbor (NN) [20], nearest subspace [38], support vector machines (SVM) [42], and sparse representation for classification (SRC) [61]. While it is possible that these classifiers may improve the accuracy of the decision, we restricted our attention to NC since the focus of this work is not on the specific classifier algorithm, but rather on the method of learning sensor locations.


This section describes our method for learning sparse sensor locations, building upon the theory described in section 2. We begin in section 3.1 by addressing classification between \( c = 2 \) categories, reducing a full measurement vector to a small handful of key measurements. Section 3.2 describes an intermediate technique that starts instead with a random subsample of the full data. In section 3.3 we develop a generalization of the algorithm to classification between \( c \geq 2 \) categories. This subsection also introduces a coupling weight \( \lambda \), which allows us to decrease the number of sensors required at the cost of slightly lower classification accuracy. Section 3.4 demonstrates the possibility of transforming the full-dimensional feature/discrimination projection into an approximate projection from the learned sparse measurements to the decision space. In most cases, it is desirable to recompute a new projection to decision space directly from the data matrix composed of sparse measurements, as in the last row of Figure 1. Finally, section 3.5 summarizes SSPOC as an algorithm.

In the following sections, we refer to features \( \Psi_r \) (e.g., PCA modes) and discrimination vectors \( w \) (e.g., LDA directions) to emphasize the generality of the method.

3.1. Deciding between two categories. Let us first consider learning sparse sensors for a classification problem between \( c = 2 \) categories. The discrimination vector \( w \) encodes the direction in \( \Psi_r \) that is most informative in discriminating between categories of observations given by \( X \). We seek a measurement vector \( s \) that satisfies \( \Psi_r^T s = w \).

In particular, we seek the sparse solution \( s \): to find the measurements that best reconstruct \( w \), our approach is to solve for

\[
(3.1) \quad \min_{s'} ||s'||_1, \quad \text{subject to} \; \Psi_r^T s' = w.
\]

The equations for \( s \) are underconstrained, so that we may use \( \ell_1 \) minimization to find the sparse solution to this convex problem. As we will show in section 4, the solution for \( s \) has at most \( r \) nonzero elements. Alternatively, it is possible to solve (3.1) for exactly \( r \) nonzero elements using a greedy algorithm [56].

It is important to remember that \( s \in \mathbb{R}^n \) and most of its entries are zero. The \( r \) nonzero elements of \( s \) represent locations of sensors that best recapture the discriminant projection vector \( \Psi_r w \).

To motivate this approach, consider that we have constructed a projection \( \Psi_r^T \) onto a set of features and a projection \( w^T \) from feature space to decision space; we
may simplify this into a single projection:

\[ \eta = w^T \Psi_r^T x_i = \langle \chi, x_i \rangle, \]

where \( \chi \triangleq \Psi_r w \). Therefore, (3.1) finds the sparse measurement vector \( s \) that projects to the same feature space coordinates as \( \chi \):

\[ \Psi_r^T s = \Psi_r^T \chi = \Psi_r^T \Psi_r w = w. \]

Finally, we see that

\[ s = \chi + \xi, \tag{3.2} \]

where \( \xi \in \ker(\Psi_r^T) \). That is to say, the sparse measurement \( s \) is the same as the vector \( \chi \) plus some residual vector that does not contain any dominant features (i.e., is not in the span of dominant features).

To compare \( \chi \) and \( s \), Figure 2 visualizes these vectors for the dog/cat recognition problem discussed in section 4.2. Figure 2(a) shows \( \chi \) for \( r = 20 \) along with the 20 pixel measurement vector \( s \) obtained by \( \ell_1 \) minimization of (3.1). The image of \( \chi \) (gray) and the sparse pixels (red) are shown in Figure 2(b). Importantly, it is not possible to obtain sparse pixel measurements by thresholding \( \chi \) (Figure 2(a)); rather, \( s \) is a sparse image that exactly projects to \( w \) in \( \Psi_r \) space.

To implement this optimal subsampling at learned sensor locations, we construct a \( q \times n \) projection matrix \( \hat{\Phi}_1 \) that maps \( x \mapsto \hat{x} \). The number of sensors \( q \) is usually equal to \( r \). The matrix \( \hat{\Phi}_1 \) are rows of the \( n \times n \) identity matrix corresponding to the nonzero elements of \( s \).
3.2. Learning from randomly subsampled data. The identical approach applies when starting from subsampled data $\tilde{X}$ (the middle “subsampled” row of Figure 1). We now solve for $\tilde{s}$ that best reconstructs the vector $\tilde{w}$ in a space defined by the columns of $\tilde{\Psi}_r$. $\tilde{\Phi}$ is then refined to construct $\hat{\Phi}_2$, the $r$ rows of the $p \times p$ identity matrix corresponding to nonzero elements of $\tilde{s}$.

Note that $\hat{\Phi}_1$ and $\hat{\Phi}_2 \tilde{\Phi}$ are both $r \times n$ matrices, where typically $r \ll p \ll n$. Although the two subsampling matrices are not the same, we show in section 4 that they are quite similar when $p/n \approx 0.1$.

Finally, having learned sparse sensor locations, we project the data into $\mathbb{R}^r$:

$$X = \hat{\Phi} X,$$

where $\hat{\Phi}$ can be either $\hat{\Phi}_1$ or $\hat{\Phi}_2 \tilde{\Phi}$, as illustrated in Figure 1. Possible methods for projecting these learned sparse measurements to decision space are described in section 3.4.

3.3. Deciding among more than two categories. The simplest extension to classification among $c > 2$ categories can be implemented by considering the projection $w^T : \mathbb{R}^r \rightarrow \mathbb{R}^{c-1}$ to decision space, followed by independently solving (3.1) for each column of $w$. However, this approach scales badly with $c$; in general, discriminating $c$ categories of data by projection into $r$-dimensional feature space results in at most $q = r(c-1)$ learned sensor locations.

An alternative formulation of the convex optimization problem solves for columns of $s \in \mathbb{R}^{n \times (c-1)}$ simultaneously; each column of $s$ projects to a column of $w$ (discrimination vector) in feature space. We introduce a norm that penalizes the total number of nonzero rows in $s$ (single measurements) to reconstruct the $c-1$ columns of $w$. Specifically,

$$s = \arg \min_{s'} \{||s'||_1 + \lambda||s' v||_1\},$$

subject to $||\Psi^T s' - w||_F \leq \varepsilon,$

(3.3)

where $||M||_1 = \sum_{ij} |m_{ij}|$, $v$ is a column vector of $(c-1)$ ones, $||M||_F = \sqrt{\sum_{ij} |m_{ij}|^2}$ is the Frobenius norm, and $\varepsilon$ is a small error tolerance ($\varepsilon \approx 10^{-10}$ for examples in section 4).

Once again, we have an underdetermined system of equations for $s$. The value of the coupling weight $\lambda$ determines the number of nonzero rows of $s$, so that the number of sensors $q$ identified is at most $\tilde{q}$ where $r \leq \tilde{q} \leq r(c-1)$.

In the limit where $\lambda = 0$, the solution to (3.3) is the same as that obtained by the uncoupled, independent approach. In general, solving (3.3) with $\lambda = 0$ would result in at most $r(c-1)$ sensor locations. As $\lambda$ becomes larger, the coupling between columns of $s$ becomes stronger, and the same pixel location can be shared between columns of $s$ to approximately reconstruct columns of $w$. In other words, the same pixel measurement is reused to capture multiple linear discriminant projection vectors. In the limit $\lambda \rightarrow \infty$, the number of sensors is bounded $r$. Following the same approach as in deciding between two categories, we implement subsampling at learned sensors to construct a projection matrix $\hat{\Phi}$.

3.4. Projecting sparse measurements to classification space. In the above procedure, we use the projections from the high-dimensional space into feature space ($\Psi^T$) and then into decision space ($w^T$) to determine the sparse sensor locations $s$. 

© 2016 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license
Using the associated sensor projection matrix $\hat{\Phi}$, it is possible to recompute the discrimination projection on the sparse data $\hat{X} = \hat{\Phi}X$. This is typically inexpensive because of the small number of rows, and this procedure is also a one-time offline cost. For multiway classification, this usually leads to better performance. The results in section 4 have all been obtained using recomputed projections from sparse measurements to discriminant space.

Even so, it is interesting that under certain circumstances, it is possible to reuse the projections $\Psi^r$ and $\omega^T$ to determine an induced projection in the decision space without retraining a new classifier. In particular, when the first $r$ PCA features $\Psi^r$ contain a significant portion of the variance in $X$, so that the truncated singular values $\sigma_{>r}$ are small, then it is possible to reuse the high-dimensional discrimination.

Given new test data $x$, we obtain a vector $\eta$ of decision space coordinates as follows:

$$\eta = \omega^T \Psi^r x. \quad (3.4)$$

We then substitute $\omega = \Psi^r \hat{s}$ into (3.4):

$$\eta = \hat{s}^T \hat{\Phi} \Psi^r x. \quad (3.5)$$

Let us define $\hat{s}$ as the measurement projection of $s$, so that $\hat{s} = \hat{\Phi} s$. Since $\hat{\Phi}$ selects the few nonzero elements of $s$, the following identity holds: $s = \hat{\Phi}^T \hat{s}$. Substituting into (3.5) yields

$$\eta = \hat{s}^T \hat{\Phi} \Psi^r x.$$  

Instead of an expression in $x$, it would be ideal to approximate $\eta$ in terms of the sparsely sampled vector $\hat{x} = \hat{\Phi} x$. This is possible if $\Psi^r \Psi^r x \approx x$, in which case $\eta \approx \hat{s}^T \hat{\Phi} x = \hat{s}^T \hat{x}$.

More precisely, let $x = x_1 + x_2$, where $x_1$ is spanned by the first $r$ PCA modes given by the columns of $\Psi^r$, and $x_2$ is the component of $x$ that is orthogonal to the columns of $\Psi^r$. The exact discrimination vector $\eta$ is given by

$$\eta = \hat{s}^T \hat{\Phi} \Psi^r \Psi^r x = \hat{s}^T \hat{\Phi} x_1 = \hat{s}^T \hat{x}_1.$$  

However, if this is approximated by $\tilde{\eta} = \hat{s}^T \hat{x}$, then there is an error associated with the orthogonal component $x_2$:

$$\tilde{\eta} = \hat{s}^T \hat{x} = \hat{s}^T (\hat{x}_1 + \hat{x}_2) = \eta + \hat{s}^T \hat{x}_2.$$  

As long as $\hat{x}_2$ is small, as is the case when the PCA projection is accurate, then the error $\hat{s}^T \hat{x}_2$ is small. In decision tasks where the training data $X$ is truly low-rank, so that the feature space $\Psi_r$ captures most of the variance in $X$, then the low-dimensional discrimination $\tilde{\eta} = \hat{s}^T \hat{x}$ may be used without retraining; this also assumes that $x$ is a representative test image that is well approximated by the column space of $X$.

In many of the examples that follow in section 4, the data $X$ is not truly low-rank, so that the first $r$ PCA modes capture only the dominant features of $X$. In these cases, we encourage a reclassification on $\hat{X}$ because of the improved accuracy and relatively low computational overhead.
3.5. The SSPOC algorithm. In practice, the SSPOC algorithm takes as inputs features \( \Psi_r \) and discrimination vectors \( w \) to learn and output a set of key sensor locations. The training data \( X \) consists of a collection of \( m \) observations, where each observation is a vector \( x_i \in \mathbb{R}^n \) and belongs to one of \( c \) distinct categories. The features \( \Psi_r \) may typically be the first \( r \) PCA modes of \( X \) so that \( \Psi_r \in \mathbb{R}^{n \times r} \), and \( w \) are typically vectors computed by LDA so that \( w \in \mathbb{R}^{r \times (c-1)} \).

The SSPOC algorithm is as follows:

1. Solve (3.3) for \( s \in \mathbb{R}^{n \times (c-1)} \). The choice of parameters is discussed below.
2. Using the sparse solution \( s \), choose \( q \) sensor locations corresponding to the \( q \) rows of \( s \) containing at least one nonzero entry. In practice, a useful threshold is to find elements of \( s \) where \( |s_{ij}| \geq \frac{||s||_F}{cr} \).
3. Construct the sparse measurement matrix \( \hat{\Phi} \in \mathbb{R}^{q \times n} \), a matrix of zeros except that each row has one entry of unity corresponding to a point sensor location.
4. Learn a new classifier using the sparsely measured data \( \hat{X} = \hat{\Phi}X \). The new classifier, which considers only the few sensors selected by SSPOC, is then used to classify new observations.

This formulation of the sensor placement optimization has two parameters, a sparsity tuning parameter \( \lambda \) and an error tolerance \( \epsilon \). For classification between two categories (\( c = 2 \)), the value of \( \lambda \) is irrelevant. When \( c > 3 \), the choice of \( \lambda \) determines the resulting sparsity in \( s \); in effect, increasing this parameter derives fewer sensor locations. The tradeoff between number of sensors and classifier accuracy is explored in section 4.3, and this tradeoff may ultimately depend on the budget for number of sensors in a particular application, given realistic cost constraints. The error tolerance \( \epsilon \) is given the value \( 10^{-10} \) for examples shown in section 4.

Functions implementing this algorithm in MATLAB, including code to reproduce the cats and dogs example, are available at https://github.com/bwbrunton/SSPOC_pub/. Our code makes use of cvx, a convex optimization solver package available at http://cvxr.com, to solve (3.3).

Alternatively, the optimization problem as formulated in (3.3) is closely related to the one solved by simultaneous orthogonal matching pursuit (S-OMP) [57, 55]. S-OMP is a greedy algorithm, so instead of a coupling weight parameter \( \lambda \), one would decide on a stopping criterion (for example, the desired number of iterations or sensors).

4. Results. We apply the SSPOC algorithm, as described in section 3.5, on five examples from different application domains as shown in Figure 3. These five datasets span a wide range of domain applications and include both static and dynamic data from computational simulation and laboratory experiments. In each experiment, we demonstrate that a classifier built on a few optimally placed sparse sensors performs comparably to a classifier built on PCA features of the full-state data. Moreover, approximately optimal sparse sensor locations may be learned from measurements randomly subsampled from 10% of the full measurements. These learned sensors cluster at intuitive locations, consistent with the coherent features in the full data.

4.1. Example: Dynamical regimes of cubic-quintic Ginzburg–Landau equation. Our first example seeks to classify dynamical regimes from solutions to the cubic-quintic Ginzburg–Landau equation (CQGLE) [36]:

\[
(4.1) \quad iq_t + \left( \frac{1}{2} - i\tau \right) q_{xx} - i\kappa q_{xxxx} + (1 - i\mu) |q|^2 q + (\nu - i\epsilon) |q|^4 q - i\gamma q = 0,
\]

where \( q(x,t) \) is a complex function of space and time.
Example 1: Dynamic regimes classification of CQGLE soln.

Example 2: Image recognition of cats and dogs

Example 3: Face recognition

Example 4: Classification of Reynolds number of flow

Example 5: Classification of cancer type

Interestingly, solutions of the CQGLE exhibit different regimes of behavior that can be characterized by the parameter values \( \beta = (\tau, \kappa, \mu, \nu, \epsilon, \gamma) \). Solutions to these equations are obtained numerically using a spectral method with a fourth-order Runge–Kutta adaptive time-stepping algorithm, simulated using 1024 Fourier modes \([37]\). Here we focus on two typical dynamical regimes, \( \beta_1 = (-0.3, -0.05, 1.45, 0, -0.1, -0.5) \) and \( \beta_2 = (0.08, -0.05, 0.6, -0.1, -0.1, -0.1) \). The solutions in these regimes are shown in Figure 3.

Fig. 3. Data from the five examples used to demonstrate learning sparse sensor placement and schematics of the sensor locations identified by the SSPOC algorithm; see section 4 for data sources and results.
To classify these two regimes, we use snapshots of the full-dimensional ($n = 1024$) solutions from both regimes stacked into columns of $X$. Figure 4(a) shows the columns of data projected on the first two principal components, along with the linear discriminant projection $w$. The top right region of this plot, where the differently shaded dots are mixed, corresponds to transient dynamics in the simulation before the attractor solution.

A histogram of the full data projected onto decision space $\eta = w^T \Psi_2^T X$ in Figure 4(b) shows the separation between the two regimes (light and dark gray bars). Applying SSPOC, we find two sensor positions that optimize the reconstruction of $w$ in $\Psi_2$ space. These two sensor locations are shown as crosses in Figure 4(c) overlaid with $\Psi_2 w$, and they fall at physically intuitive locations. The first sensor measures the center peak, the second sensor measures one of the side lobes, and these two sensors—out of 1024 possible sensors in the full measurement space—separate the two dynamic regimes (Figure 4(d)). In this example, using full data and two-sensor data produces classification accuracy of 0.944 and 0.994, respectively.

4.2. Experiment: Cats and dogs. We seek to classify images of cats versus images of dogs. As shown in Figure 3, each image of a pet face belongs to a distinct species, and there is considerable variability between individuals of the same species. Notably, members of each species take on a large range of colorations in fur, markings, and ear postures. We used 121 images each of cats and dogs; images have $n = 64 \times 64 = 4096$ pixels and are stacked to form columns of $X$.

We compared classification accuracy using learned measurements versus (1) using the principal components of the full image and (2) using random pixel measurements. The accuracy for all of these methods improved with larger numbers of sensors/features; $r$ learned sensors performed almost as well as $r$ principal components and consistently better than $r$ random pixel sensors. Figure 5 shows the results of experiments where classifiers were trained on a random 90% of the images and assessed on the remaining 10% of images. The mean and standard deviation of the cross-validated accuracy over 400 random iterations of training/test images are plotted. Classifiers built on less than 1% of the total pixels (green and red lines in Figure 5(a)) performed nearly as well as projections to principal components of the full image.
Accuracy reached a plateau at around $r = 15$ features/sensors (Figure 5(b)) of about 80%, suggesting that the discriminant vector between cats and dogs may be overfit when more than 15 features are used. Figure 5(b) shows mean classification performance using $r$ sensors learned from $p$ subsampled measurements and $r$ features, where $p$ and $r$ are varied systematically. Sparse sensors learned from $p = 400$ pixels did almost as well as those learned from $p = n = 4096$ pixels. In other words, we were able to learn a nearly optimally sparse set of sensors starting from an already massively undersampled set of pixels.

The performance of all of these approaches is limited by the large variability in appearances within the categories of cats and dogs. In fact, the images that are most often misclassified include animals that are predominantly one color (black or white) and dogs with pointy as opposed to droopy ears. In other words, some dog images may not lie close to the bulk of the other dog images in feature space, so that a linear classifier is not able to capture the complex geometry separating the two categories.

It is possible to build classifiers from random pixels that perform significantly better than chance. The improvement in accuracy of using computed features or learned sensors over random pixels became smaller as $r$ increased (Figure 5(a)). The perhaps surprising performance of random pixels is consistent with the notion of enhanced sparsity and the observation of Wright et al. [61] that using random features, as long as there are enough of them, serves classification as well as using engineered features.

If the sparse sensors represent pixels that are key to the decision, then they should be clustered at locations that are maximally informative about the difference between cats and dogs. Indeed, an average of sensor locations learned over 400 random iterations shows clustering around the animals’ mouth, forehead, eyes, and tops of ears (Figure 6(b)). When sparse sensors were learned from an already subsampled dataset, a qualitatively similar distribution of sensor locations was found, where each cluster of sensors showed a slightly larger spatial variance. Such an ensemble of sensor placements can be thought of as a mask for facial features particularly relevant for the classification.

Our algorithm applies equally well when the subsampled dataset (middle row in
Figure 6. Classification accuracy of sparse sensors learned from random projections (black dashed line in (a)) is the same as sensors learned from single pixels (solid red line in (a)). Even so, ensemble of sparse sensor locations learned from single pixels (b) and from random projections (c) reveals that using random projections may be a poor engineering choice. While sensor locations in (b) show a tendency to measure at the eyes, nose, mouth, and forehead, sensor locations in (c) do not cluster at the visually salient features in the data.

Figure 1) is not a set of random pixels but a set of random projections. Figure 6 shows that classifiers built on $r$ sensors learned from $p = 400$ pixels or $p = 400$ random projections perform identically. Random projections used to produce this result were composed of ensembles of Bernoulli random variables with mean of 0.5. Even so, Figure 6(c) makes clear that using random projections instead of pixels is a poor engineering choice. The ensemble of sensors computed from random projections does not illuminate coherent facial features.

4.3. Example: Extended Yale Face Database B. We extended sparse sensor learning to classification between more than two categories, applying our approach to human face recognition. We used the Extended Yale Face Database B [26, 38], which contains images of 28 individual faces captured under various lighting conditions (see Figure 3). Images have been previously aligned and cropped; each image has $n = 192 \times 168 = 32,256$ pixels, and each individual has $m = 64$ images. Comparing the faces dataset with the cat/dog dataset in section 4.2, there is significantly less variability within each category in facial features, although large portions of the faces were effectively occluded by lack of illumination.

We demonstrated our sensor learning approach to learn sensor locations that categorize $c = 3, 4$, and 5 faces and compared the classification accuracy of learned sensors to projections to PCA features of the full image and to random sensors of the same number. Figure 7 shows that as the coupling weight $\lambda$ is increased, the number of learned sensors decreases. When $\lambda = 0$, an upper bound of $r(c - 1)$ pixel locations was found. In the limit $\lambda \to \infty$, the number of sensors was bounded by $r$, the number of PCA features included in the discrimination. By adjusting the magnitude of $\lambda$, we gain control over the number of sensor locations used in the classification. When individual sensors are expensive, it may be desirable to use fewer sensors at the cost of slightly lower performance.

The cross-validated accuracy shown in the bottom row of Figure 7 represents results obtained from recomputing the LDA projection $\hat{w}^T$ to decision space after learning the sensor locations. This strategy is not typically an expensive computation, as the number of rows in the measurement vectors is small. Further, an LDA classifier tailor-made for the specific learned pixels can outperform the PCA approach when the number of sensors exceeds the number of PCA features. In contrast, the induced
Fig. 7. Classification of 3, 4, and 5 faces as the coupling weight $\lambda$ varies between 0 and 100. Number of sensors and the cross-validated performance are shown, comparing sensors learned with no subsampling (solid green lines), sensors learned from 10% of the pixels (solid red lines), and random sensors of the same number (dashed blue lines) against the accuracy achieved by using $r$ PCA features of the full images (blue square). Each instance of the classifier was trained on a random 75% of the images and evaluated on the remaining 25%; error bars are standard deviations.

Fig. 8. Increasing the coupling weight $\lambda$ results in fewer sensor locations that capture approximately the same features. In both panels, the underlying image is a visualization of the decision vector $\chi = \Psi \hat{w}$ for an example of categorizing $c = 3$ faces using $r = 10$ features. Panel (a) shows the locations of sparse sensors that independently reconstruct the $c - 1 = 2$ columns of $w$ (yellow and green dots, 20 total). The sensors boxed in red are aggregated in panel (b), where the coupling weight $\lambda$ brought the total number of sensors down to 15.

projection method described in section 3.4 cannot perform better than the PCA-LDA approach. For classification among $c > 2$ categories, especially for larger $\lambda$ coupling weights, recomputing the LDA projection usually leads to more accurate results.

Interestingly, sensors learned from randomly subsampled pixels (10% of the original pixels) performed exactly as well as sensors learned from the full image (red and green lines in Figure 7). Obtaining sparse sensors from already subsampled data presents significant savings in the learning procedure, both in the SVD to extract a feature space and in the convex optimization to solve for a sparse $s$.

Figure 8 illustrates an example of increasing $\lambda$ on the number and locations of sensors identified by the solution to the optimization in (3.3). For this example, $c = 3$ faces were in the training set, so $w$ and $s$ both have $c - 1 = 2$ columns. When $\lambda = 0$, ...
the columns of $\mathbf{w}$ (linear discriminant vectors in $r = 10$ dimensional PCA feature space) are treated independently, and $r(c - 1) = 20$ total sensors are found. Notice, however, in Figure 8(a) that certain pairs of sensor locations are in close proximity (red boxes) and likely carry information about the same nonlocal facial feature. As $\lambda$ increases and the total number of sensors is penalized, these sensor pairs appear to collapse onto single sensors in Figure 8(b). Comparing the two sets of sensor locations, we can see that, in addition to aggregated sensors, some sensors have remained the same (for example, the tops of each eyebrow), some sensors have disappeared (on the cheek, at bottom right), and some entirely new sensors have appeared (lower right corner of eye).

Sparse sensors for face recognition cluster at major facial features: the eyes, the nose, and corners of the mouth (Figure 9). The nose is particularly prominent in these masks, consistent with the fact that due to the eccentricity in illumination, the nose is the only facial feature reliably visible in all images. Interestingly, this data-driven algorithm identifies the same features that are favored by human gaze. As first noted by Yarbus [62], humans examining an image of a face spend a preponderance of time fixating on the eyes, the nose, and the mouth.

4.4. Example: Reynolds number of flow past a cylinder. This example seeks to classify the Reynolds number of flow past a cylinder [43] using pressure measurements on the surface of the cylinder. Three different Reynolds number flow regimes were used in the data: $Re = 40, 150, 1000$. These correspond to the flow becoming more complex, for example, by increasing the flow velocity. At $Re = 40$, the wake behind the cylinder corresponds to a steady, stably attached laminar separation bubble, although this bubble breaks at around $Re = 47$, followed by periodic vortex shedding. As the Reynolds number increases, additional instabilities develop, resulting in increasingly complex spatial-temporal structures in the wake. Although three-dimensional instabilities typically develop before $Re = 1000$, this example has been useful for demonstrating sparse sampling strategies in fluid dynamics [5].
Fig. 10. Classification of Reynolds number of flow past a cylinder, using three different Reynolds number regimes. (a) Classification accuracy using \( r \) sensors increases with \( r \), as computed by cross-validation. In this case, the magnitude of the noise is 0.1 (10\% of magnitude of signal). In contrast, the dashed black line is the accuracy if the same number of sensors were placed randomly. (b) Cross-validated accuracy as the magnitude of noise increases. Error bars are standard deviations.

The time course of pressures measured at these Reynolds numbers is shown in Figure 3. For each regime, the full-state data consisted of 269 pressure measurements collected at 100 time points. Measurements were equally spaced on the surface of the cylinder body as computed by flow simulations described in detail previously \cite{5}. The entire cylinder flow dataset of \( c = 3 \) classes consisted of 300 samples at \( n = 269 \) measurement locations. For the classification results shown here, Gaussian measurement noise was added to the pressure measurements.

Figure 10(a) shows the cross-validated accuracy of the classification for Gaussian noise of 10\% magnitude as \( r \), the number of sensors, increases. Using 5 optimally placed sensors, SSPOC achieves a mean accuracy of 95\%. Sparsely placed sensors outperform randomly placed sensors of the same number by approximately 10\% for number of sensors less than 6. Because of the inherent low dimensionality of this dataset, using PCA-LDA with the full-state data produces classification with close to zero error.

Figure 10(b) shows the cross-validated accuracy of 5 SSPOC sensors as the magnitude of Gaussian noise is increased up to 30\%. As before, these key measurements consistently outperform randomly placed measurements. Figure 11 is a visualization of where these key sensor locations are distributed on the cylinder body. Each panel shows the distribution of sensors over 100 random cross-validated iterations, as in Figure 10(b). Sensor locations that are more likely to be chosen over the random iterations are larger, darker circles. Over all noise magnitudes, there are three regions of the cylinder that are favored for classification of the Reynolds number of the
Fig. 12. Cancer classification by gene expression profile as measured by a cDNA microarray. (a) Accuracy of sensor classification from SSPOC is shown in blue and red for cross-validated replicates and validation test samples, respectively. For comparison, accuracy of randomly chosen sensors of the same number is shown as the black dashed line. (b) The 28 genes identified by SSPOC and their expression profiles for the 63 samples. Genes are labeled by their IDs as given in the dataset. The samples are sorted by tumor type. This profile corresponds to the \( r = 28 \) case marked by the red triangle in (a) and performs at 100% accuracy. Dark blue is low gene expression, and yellow is high expression. Many of these genes overlap with the set of 43 identified by the shrunken centroids method [54].

flow, two at the leading half of the cylinder and one at the back. In our dataset, the two regions in the leading half are not symmetric because of an asymmetry in the raw pressure measurement data we used. These regions are physically interpretable because they correspond to regions on the cylinder where pressure differences among the three different classes of flows are the largest.

4.5. Example: Cancer classification by microarrays. We applied SSPOC to classifying tumor types using the expression profiles of a few genes in a microarray. Specifically, we used data from measuring expression levels of 6567 genes from cDNA microarrays of 83 samples of small round blue cell tumors (SRBCT) of childhood [34]. The SRBCT tumors were identified as one of four \((c = 4)\) types: rhabdomyosarcoma (RMS), Ewing sarcoma (EWS), neuroblastoma (NB), or Burkitt lymphoma (BL). The full dataset of gene expression profiles of each sample is shown as a heat map in Figure 3. This dataset or a subset of it had previously been analyzed using neural network and statistical methods [34, 54].

This microarray dataset is distinct from the previous examples in part because here the measurement “locations” are genes and have no relevant physical relationship to each other. Each sensor is a probe that measures the expression level of a gene, so the sparse sensor locations identified should be interpreted as the few genes whose expression profiles best differentiate the four tumor types.

In Figure 12(a), we show the accuracy of the tumor classification as the number of sensors increases. For these analyses, the coupling weight \( \lambda = 100 \), so that training on \( r \) PCA features produced at most \( r \) sensor locations (see section 3.3). In blue are the cross-validated accuracies using the first 63 tumor samples, computed using 100 replicates where a random 20% of these samples were withheld as the test set. In addition, for each \( r \), we used the distribution of sensors identified by the cross-validated training to identify a consensus set of \( r \) sensors, and then used the classifier
learned from these $r$ sensors to classify the last 20 tumor samples. The classification accuracy of these 20 test samples is shown as the red curve in Figure 12(a). Classifiers based on sensors identified by SSPOC (blue and red lines) perform significantly better than random sensors of the same numbers (black dashed line). In fact, very close to 100% accuracy on the test samples is possible with 25–35 genes identified with SSPOC.

Looking more closely at the gene sensor locations identified by SSPOC, Figure 12(b) shows the expression profiles of 28 genes for all 83 samples. The samples are arranged by tumor type, and the genes are ordered horizontally by results of k-means clustering. These 28 genes are able to classify the 20 test samples with 100% accuracy. This number of sensors compares with the 43 genes chosen by the shrunk centroids method [54] and 96 genes identified by the neural network method [34] to achieve comparable accuracy. Our 28 genes have 8 genes in common with the 43 identified by Tibshirani et al. [54]. It is worth noting that their analysis considered a subset (2308/6567) of the genes in the full microarray dataset, and only 16/28 of our gene sensors fall within this subset.

5. Discussion. In this paper, we described the sparse sensor placement optimization for classification (SSPOC) algorithm that refines a large set of measurement locations to learn a much smaller subset of key locations to best serve a classification task. The algorithm exploits enhanced sparsity for classification, for applications where reconstruction of the full state may be bypassed. Enhanced sparsity provides an orders-of-magnitude reduction in the number of measurements required when compared with standard compressed sensing strategies. Measurements are projected directly into decision space; reconstruction from so few measurements is neither possible nor needed.

Our algorithm leverages the sparsity promoting $\ell_1$ minimization as phrased in (3.1) or (3.3). These convex optimization problems solve for a sparse set of sensor locations to closely approximate the linear discrimination vector in PCA space. The framework of SSPOC may be generalized to a variety of feature extraction algorithms. For instance, where the separation between different categories is not linear, nonlinear transformations may be applied to extract relevant features in the dataset. In addition, once a set of sensors has been obtained, a number of more sophisticated classifier algorithms may be applied to these measurements to improve the classification accuracy.

We demonstrated our approach on five examples of categorization tasks from different application domains. Classifiers built on learned sensors approached the performance of classifiers built on projections to principal components of the full-state data. These optimal sensor locations could be learned approximately even from already randomly subsampled data. In either case, the learned sensors performed significantly better than the matched number of randomly chosen sensors. Furthermore, the ensemble of learned sensor locations clustered around coherent features of the data.

It is plausible that biological organisms may exploit enhanced sparsity for classification. Organisms interact with the external world with motor outputs, which are often discrete trajectories at specific moments in time, so that the transformation from sensory inputs to motor outputs can be thought of as a classification task. For example, a fly has no need to reconstruct the full flow velocity field around its body—it has only to decide what to do in response to a gust. Sensory organs and data processing by the nervous system can be expensive; therefore, it is advantageous to place a smaller number of sensors at key locations on the body. Beyond
those demonstrated in this work, other potential applications of SSPOC include the detection of disease spread in epidemiological monitoring, mobile sensor networks in oceanographic and atmospheric sampling, network behaviors in the electrical grid, internet packet routing, and transportation.

Acknowledgments. We especially thank Tom Daniel and Leifeng Bo for helpful discussions related to this work, and Anna Gilbert for insightful comments on an earlier version of the manuscript.

REFERENCES

SPARSE SENSORS FOR CLASSIFICATION


© 2016 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license


© 2016 SIAM. Published by SIAM under the terms of the Creative Commons 4.0 license