USING THE KERNEL TRICK IN COMPRESSIVE SENSING: ACCURATE SIGNAL RECOVERY FROM FEWER MEASUREMENTS

Hanchao Qi, Shannon Hughes
Department of Electrical, Computer, and Energy Engineering
University of Colorado at Boulder

ABSTRACT

Compressive sensing accurately reconstructs a signal that is sparse in some basis from measurements, generally consisting of the signal’s inner products with Gaussian random vectors. The number of measurements needed is based on the sparsity of the signal, allowing for signal recovery from far fewer measurements than is required by the traditional Shannon sampling theorem. In this paper, we show how to apply the kernel trick, popular in machine learning, to adapt compressive sensing to a different type of sparsity. We consider a signal to be “nonlinearly K-sparse” if the signal can be recovered as a nonlinear function of K underlying parameters. Images that lie along a low-dimensional manifold are good examples of this type of nonlinear sparsity. It has been shown that natural images are as well [1]. We show how to accurately recover these nonlinearly K-sparse signals from approximately 2K measurements, which is often far lower than the number of measurements usually required under the assumption of sparsity in an orthonormal basis (e.g. wavelets). In experimental results, we find that we can recover images far better for small numbers of compressive sensing measurements, sometimes reducing the mean square error (MSE) of the recovered image by an order of magnitude or more, with little computation. A bound on the error of our recovered signal is also proved.

Index Terms—Compressive sensing, Kernel methods.

1. INTRODUCTION

The “kernel trick” in machine learning is a way to easily adapt linear algorithms to nonlinear situations. For example, by applying the kernel trick to the support vector machines (SVM) algorithm, which constructs the best linear hyperplane separating data points belonging to two different classes, we obtain the kernel SVM algorithm, an algorithm that constructs the best curved boundary separating data points belonging to two different classes. Similarly, principal components analysis (PCA), selects the best linear projection of the data to minimize error between the original and projected data. Kernel PCA finds the best smooth polynomial mapping to represent data.

The key idea of the kernel trick is that, conceptually, we map our data from the original data space \( \mathbb{R}^n \) to a much higher-dimensional feature space \( \mathcal{F} \) using the nonlinear mapping \( \Phi : \mathbb{R}^n \rightarrow \mathcal{F} \) before applying the usual linear algorithm such as SVM or PCA in the feature space. As an example, we might map a point \((x_1, x_2) \in \mathbb{R}^2\) onto the higher-dimensional vector with components \(x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_2^3, \ldots\) before applying SVM or PCA. A linear boundary in the higher-dimensional feature space \(\sum_{i} a_i \Phi(x)_i = C\) can be expressed as a polynomial boundary in the original space \(a_0 x_1 + a_1 x_2 + a_2 x_1^2 + \ldots = C\). Similarly, a linear mapping of the data becomes a polynomial mapping.

However, this view of the kernel trick is purely conceptual. In reality, we avoid the complexity of mapping to and working in the high-dimensional feature space. When the original algorithm can be written in terms of only inner products between points, not the points themselves, we can replace the original inner product \(\langle x, y \rangle\) with the new inner product \(k(x, y) = \langle \Phi(x), \Phi(y) \rangle\) and run the original algorithm without additional computation. For example, a popular choice of \(k(x, y)\) is \((\langle x, y \rangle + c)^d\), which produces a \(\Phi\) of monomials as described above. As an illustration, for \(x, y \in \mathbb{R}^2\), \(c = 0\), \(d = 2\), \(k(x, y) = \langle x, y \rangle^2 = \langle x_1^2, \sqrt{2} x_1 x_2, x_2^2 \rangle, \langle y_1^2, \sqrt{2} y_1 y_2, y_2^2 \rangle\), so \(\Phi(x) = (x_1^2, \sqrt{2} x_1 x_2, x_2^2)\).

Kernel PCA [2] often reveals low-dimensional representations of the dataset that reflect its underlying degrees of freedom. For example, in the synthetic “sculpture faces” dataset of Fig. 1, each face image is a highly nonlinear, but deterministic, function of three underlying variables: two pose angles and one lighting angle. A kernel PCA, performed with a well-chosen kernel function, is able to pick out two of these degrees of freedom as the first two dimensions chosen in kernel PCA. (Note that an ordinary PCA will not.) The results of kernel PCA can thus reflect a type of nonlinear sparsity in the dataset. We could represent each image fairly accurately, knowing only its coordinates in this two-dimensional representation.

Indeed, we may be able to build a better approximation of the image knowing its first \(m\) coordinates in a nonlinearly sparse representation such as kernel PCA than we can knowing its largest \(m\) Fourier, wavelet or curvelet coefficients. Fig. 1(c) shows a comparison of the mean-squared error for an individual image of the “sculpture faces” dataset when approximated from \(m\) kernel PCA components vs. \(m\) wavelet coefficients. The MSE decays much faster for kernel PCA components, showing that the image is more nonlinearly sparse than linearly sparse. Like this simple toy dataset, natural images have been shown to be nonlinearly sparse: patches of natural images tend to lie along low-dimensional manifolds [1].

In view of this nonlinear sparsity, consider compressive sensing [3, 4]. Recently, compressive sensing has asserted that we can achieve perfect reconstruction of a signal with far fewer samples than Shannon-Nyquist traditionally requires, if the signal is approximately sparse in some basis. In fact, in practice, we can achieve a near-perfect, or even perfect, reconstruction, of the signal from about \(5K\) measurements, each of which is the \(K\)-sparse signal’s inner product with a random vector. In this paper, we will show how the kernel trick can be used to adapt this paradigm of reconstructing a linearly sparse signal from a linear set of measurements to the case of reconstructing a nonlinearly sparse signal from either nonlinear or linear measurements. The key idea is that a signal that is nonlinearly sparse can, with a proper choice of kernel, become linearly sparse in feature space, as our “sculpture faces” did above. We can thus reconstruct it from random measurements in feature space, which can be easily obtained from the usual random measurements for some kernels. Experimentally, we find that when the signal to be reconstructed is nonlinearly sparse, our method reconstructs it from far fewer compressive sensing measurements, sometimes using an order of magnitude fewer measurements to achieve the same MSE.

Section 2 outlines our recovery algorithm. Section 3 presents experimental results showing its power on sample datasets. Finally,
in Section 4, we present some error analysis for our recovered signal.

2. SIGNAL RECOVERY FROM CS MEASUREMENTS USING THE KERNEL TRICK

Suppose we wish to recover an unknown \( y \) that is approximately nonlinellarly \( d \)-sparse, meaning it can be reconstructed with small error from its first \( d \) coordinates in some orthonormal basis of feature space, i.e.

\[
\Phi(y) \approx \sum_{k=1}^{d} \beta_k v_k. 
\]

Here, unlike traditional compressive sensing, we do not expect the \( v_k \) to be some unknown subset of the standard basis for \( \mathcal{F} \), which are typically monomials. Hence, \( \{v_k\}_{k=1}^{d} \) in \( \mathcal{F} \) will typically be estimated via kernel PCA from other data that is expected to be nonlinearly sparse in the same way, i.e., from other samples \( \{x^{(i)}\}_{i=1}^{m} \) of the manifold of images that our image belongs to, or from other natural images.

Now suppose we have measurements of \( y \) in the form of linear inner products \( \langle y, e_i \rangle \), or nonlinear inner products \( k(y, e_i) = \langle \Phi(y), \Phi(e_i) \rangle \), where \( \{e_i\}_{i=1}^{n} \) are known random vectors. Our goal is to reconstruct \( y \) from the random measurements as in typical compressive sensing. In the case that linear inner products are provided, we shall assume the kernel defining our feature space is in the form \( f(y, e_i) \) so that \( k(y, e_i) \) is known as well.

If we complete the orthonormal basis with \( v_{d+1}, \ldots, v_q \), where \( q \) is the dimension of \( \mathcal{F} \), we can write \( \Phi(y) \) as

\[
\Phi(y) = \sum_{k=1}^{d} \beta_k v_k + \sum_{k=d+1}^{q} \gamma_k v_k. 
\]

Each random measurement can then be expressed as

\[
k(y, e_i) = \langle \Phi(y), \Phi(e_i) \rangle = \sum_{k=1}^{d} \beta_k \langle v_k, \Phi(e_i) \rangle + \epsilon_i
\]

where \( \epsilon_i \) is a small error term since \( \sum_{k=d+1}^{q} \gamma_k v_k \) is small by assumption. In section 4, we will analyze the error incurred by this.

In matrix form, this becomes

\[
\begin{bmatrix}
k(y_1, e_1) \\
k(y_2, e_2) \\
\vdots \\
k(y_n, e_n)
\end{bmatrix} =
\begin{bmatrix}
\langle v_1, \Phi(e_1) \rangle & \langle v_2, \Phi(e_1) \rangle & \ldots & \langle v_d, \Phi(e_1) \rangle \\
\langle v_1, \Phi(e_2) \rangle & \langle v_2, \Phi(e_2) \rangle & \ldots & \langle v_d, \Phi(e_2) \rangle \\
\vdots & \vdots & \ddots & \vdots \\
\langle v_1, \Phi(e_n) \rangle & \langle v_2, \Phi(e_n) \rangle & \ldots & \langle v_d, \Phi(e_n) \rangle
\end{bmatrix}
\begin{bmatrix}
\beta_1 \\
\beta_2 \\
\vdots \\
\beta_d
\end{bmatrix}
+ 
\begin{bmatrix}
\epsilon_1 \\
\epsilon_2 \\
\vdots \\
\epsilon_n
\end{bmatrix}
\]

Writing the above equation as

\[
M = G \beta + \epsilon,
\]

we can use the least squares estimator to estimate \( \beta \): \( \hat{\beta} = G^+ M \). We shall later show that we must have \( n \geq d \) for good results, so we will always have \( G^+ = (G^T G)^{-1} G^T \).

We then estimate \( \Phi(y) \) as \( \hat{\Phi}(y) = \sum_{k=1}^{d} \hat{\beta}_k v_k \). Finally, we need to invert the one-to-one mapping \( \Phi \) to find our estimate \( \hat{y} \) of \( y \). This problem, of recovering \( z \) from \( \Phi(z) \), is called the preimage problem in the kernel methods literature. If an exact pre-image \( z \) such that \( \Phi(z) = \Phi(y) \) exists, then we can estimate our original signal \( y \) via \( \hat{y} = \Phi^{-1}(\hat{\Phi}(y)) \).

However, more often due to estimation error in \( \hat{\Phi}(y) \), an exact preimage will not exist, and we will set \( \hat{y} = \arg \min_{\hat{z}} ||\Phi(\hat{z}) - \hat{\Phi}(y)||^2 \).

2.1. Avoiding Computation in Feature Space

Often, we want to avoid working with \( \Phi \) or elements in \( \mathcal{F} \), such as \( \{v_k\}_{k=1}^{d} \), to avoid complexity. We now show that our estimate from the previous subsection can be obtained without working in \( \mathcal{F} \) if we rely on the data \( \{x^{(i)}\}_{i=1}^{m} \) from which \( \{v_k\}_{k=1}^{d} \) were estimated.

In the following description, for simplicity, we assume that the mapped data \( \Phi(x^{(j)}) \) are centered in \( \mathcal{F} \). Otherwise, we must replace \( \Phi(x^{(j)}) \) with the centered \( \tilde{\Phi}(x^{(j)}) = \Phi(x^{(j)}) - \frac{1}{m} \sum_{i=1}^{m} \Phi(x^{(i)}) \).

In the usual kernel PCA procedure (see [2]), we recover not the \( v_k \)s themselves, which are cumbersome, but rather their coefficients \( \alpha_k \) in the expansion

\[
v_k = \sum_{i=1}^{m} \alpha_k^i \Phi(x^{(i)})
\]

(It can be easily shown that the optimal \( v_k \) for PCA must lie in the subspace spanned by the \( \Phi(x^{(i)}) \)). This allows us to express \( \langle v_i, \Phi(e_i) \rangle \) as \( \sum_{j=1}^{d} \alpha_j^i k(x^{(j)}, e_i) \) and thus \( G \) as

\[
G = \begin{bmatrix}
k(x^{(1)}, e_1) & \ldots & k(x^{(m)}, e_1) \\
\vdots & \ddots & \vdots \\
k(x^{(1)}, e_n) & \ldots & k(x^{(m)}, e_n)
\end{bmatrix}
\begin{bmatrix}
\alpha_1^1 & \ldots & \alpha_1^d \\
\vdots & \ddots & \vdots \\
\alpha_m^1 & \ldots & \alpha_m^d
\end{bmatrix}
\]

Once we have found \( \hat{\beta} = G^+ M \), we then have

\[
\hat{\Phi}(y) = \sum_{k=1}^{d} \hat{\beta}_k v_k = \sum_{i=1}^{m} \sum_{k=1}^{d} \hat{\beta}_k \alpha_k^i \Phi(x^{(i)}),
\]

We can recover the exact preimage from \( \hat{\beta} \) without venturing into feature space using the following technique in [5]. Consider an expansion \( \Psi = \sum_{i=1}^{m} c_i \Phi(x^{(i)}) \) in \( \mathcal{F} \). If there exists \( z \in \mathbb{R}^p \) such
that $\Phi(\mathbf{z}) = \Psi$, and an invertible function $f_k$ such that $k(\mathbf{x}, \mathbf{y}) = f_k(\langle \mathbf{x}, \mathbf{y} \rangle)$, then

$$
\mathbf{z} = \sum_{i=1}^{p} \langle \mathbf{z}, \mathbf{u}_i \rangle \mathbf{u}_i = \sum_{i=1}^{p} f_k^{-1} \left( \sum_{j=1}^{m} c_j \langle \mathbf{u}_j, \mathbf{u}_i \rangle \right) \mathbf{u}_i
$$

(4)

where $\{ \mathbf{u}_i \}_{i=1}^{p}$ is any orthonormal basis of $\mathbb{R}^p$. As an example kernel, we could choose $k(\mathbf{x}, \mathbf{y}) = f_k((\mathbf{x}, \mathbf{y})) = (\langle \mathbf{x}, \mathbf{y} \rangle + c)^d$, with $d$ odd. Approximate preimages can also be constructed without venturing into feature space (for example, see [5] and [6]). However, we will not detail these methods here since we found that the simple preimage recovery method in Eq. 4 was sufficient.

![Experimental Results: A visual results comparison (left) and plots (in log-log scale) of MSE vs. number of measurements (right) for each of the “sculpture face” (top), “Frey face” (middle), and “Handwritten digits” (bottom) datasets. Our method (KTCS) is compared with traditional L1 and Total Variation Minimization (TVM).](image)

**3. EXPERIMENTAL RESULTS**

In this section, we apply our method to sample signals to compare it with existing compressive sensing signal recovery techniques. For our analysis, we choose three datasets in which the images show the type of “nonlinear sparsity” previously described. The “sculpture face” dataset includes 698 images with size $64 \times 64$ of a sculpture face rendered according to 3 different input parameters, two of pose angle and one of lighting angle. The “Frey face” dataset includes 1964 images with size $32 \times 32$ of the same person’s face as he shows different emotions. The “Handwritten digits” dataset includes 60000 training images with size $32 \times 32$ of digits from 0 to 9.

All experiments were done using the kernel $k(\mathbf{x}, \mathbf{y}) = ((\mathbf{x}, \mathbf{y}) + c)^d$, setting $c$ as 0.5 times the mean of all entries of the covariance matrix formed from the points $\{ \mathbf{x}^{(i)} \}_{i=1}^{d}$, so that it is of approximately the same scale as the inner products, and for i.i.d. $\mathbf{e}_k \sim \mathcal{N}(0, \frac{1}{n} I)$. To mimic a real-world situation, we estimate $\{ \mathbf{e}_k \}_{k=1}^{d}$ via kernel PCA on the other data samples, choosing $d = \frac{n}{2}$, and avoid the complexity of feature space computation as in Sec. 2.1.

Fig. 2 compares our method (KTCS) with the popular compressive sensing recovery methods L1-Minimization (L1-MIN) and Total Variation Minimization (TVM), giving the reconstructed images for varying numbers of measurements and plotting MSE vs. number of measurements in logarithmic scale for each method. We notice that our method outperforms traditional compressive sensing recovery methods for small numbers of measurements, producing either a greatly reduced MSE for the same number of measurements or vice versa. L1-MIN and TVM do catch up and surpass our method for very large numbers of measurements and very small recovery error, most likely due to small inaccuracies in estimating the pre-image from $\Phi(y)$. These inaccuracies should decrease with exact knowledge of $\{ \mathbf{v}_i \}_{i=1}^{d}$ or an increased set of data $\{ \mathbf{x}^{(i)} \}_{i=1}^{m}$ from which to estimate them. Even though these are simple datasets, this serves as a proof-of-concept that nonlinearly sparse signals can be recovered remarkably well from very few measurements, and it suggests that more accurate recovery should also be possible for other nonlinearly sparse signals, such as natural images.

Finally, since real-world signals are not exactly sparse, but rather show a rapidly decaying approximation error with increasing number of components kept, Fig. 3 attempts to experimentally find the optimal level of nonlinear sparsity to assume for a given number of measurements $n$. It plots the MSE between the original image and our reconstructed image for different combinations of assumed sparsity level $d$ and number of measurements $n$ for the “sculpture faces” dataset. In Fig. 3(a), we see a sharp drop in MSE for each curve, occurring precisely when $n$ becomes larger than $d$. We shall understand this drop better using our error analysis in Section 4. Given $n > d$, we obtain a smaller MSE for larger $d$. In Fig. 3(b), for each $n$, the MSE is smallest at about $d \approx \frac{n}{2}$. Based on this guideline, we chose $d = \frac{n}{2}$ in our experiments.

**4. ERROR ANALYSIS OF OUR ESTIMATOR**

**Theorem.** Suppose $\mathbf{v}_1, \ldots, \mathbf{v}_d$ is an orthonormal basis for the feature space $\mathcal{F}$. Now consider $\Phi(\mathbf{y}) \in \mathcal{F}$ and its representation in terms of this basis:

$$
\Phi(\mathbf{y}) = \sum_{i=1}^{d} \beta_i \mathbf{v}_i + \sum_{j=d+1}^{m} \gamma_{j-d} \mathbf{v}_j
$$

(5)

Suppose we take $n$ measurements of $\Phi(\mathbf{y})$ by taking inner products $M_n = \{ \langle \mathbf{e}_k, \Phi(\mathbf{y}) \rangle \}$ for vectors $\{ \mathbf{e}_k \}_{k=1}^{d}$ in feature space $\mathcal{F}$ that are i.i.d drawn from an isotropic Gaussian distribution $\mathbf{e}_k \sim \mathcal{N}(0, \frac{1}{n} I)$. Let $\Phi(\mathbf{y}) = \sum_{j=1}^{d} \beta_j \mathbf{v}_j$ for $\beta = \mathcal{N}(0, \frac{1}{n} I)$.

![Experimental Results: A visual results comparison (left) and plots (in log-log scale) of MSE vs. number of measurements (right) for each of the “sculpture face” (top), “Frey face” (middle), and “Handwritten digits” (bottom) datasets. Our method (KTCS) is compared with traditional L1 and Total Variation Minimization (TVM).](image)

$$
\left\| \Phi(\mathbf{y}) - \hat{\Phi}(\mathbf{y}) \right\|^2 \leq \frac{\left\| \beta \right\|_2^2 + b}{\left( 1 - \sqrt{d/n - r} \right)^2} + \left\| \gamma \right\|^2
$$

(6)

with probability at least $\frac{\beta^2}{\beta^2 + \frac{1}{n} \left\| \gamma \right\|^2} (1 - e^{-nr^2/2}).$
We note that because of the rapid decay of error with number of kernel PCA components for real signals, as displayed in Fig. 1, \( \| \gamma \|_2^2 \) should decay rapidly for increasing \( d \), becoming very small and ensuring a tight bound. Also, as \( n \to \infty \), we find that \( \| \Phi(y) - \hat{\Phi}(y) \|_2^2 \leq 2\|\gamma\|_2^2 \) with prob. 1.

Proof. The measurement vector \( M \) can be written as

\[
M_{n\times d} = G_{n\times d} \beta_{d\times 1} + H_{n\times (q-d)} \gamma_{(q-d)\times 1}
\]

where the subscripts indicate dimensions and \( H \) is the matrix with \( i,j \)th entry \( H_{ij} = \langle e_i, v_{j+d} \rangle \). The error is thus

\[
\| \Phi(y) - \hat{\Phi}(y) \|_2^2 = \left\| \sum_{i=1}^{d}(\beta_i - \hat{\beta}_i)v_i + \sum_{j=d+1}^{q} \gamma_{j-d}v_j \right\|_2^2 \\
= \| \beta - \hat{\beta} \|_2^2 + \| \gamma \|_2^2 (7)
\]

where

\[
\| \beta - \hat{\beta} \|_2 = \| \beta - G^*M \|_2 \\
= \| \beta - G^*(G\beta + H\gamma) \|_2 \\
\leq \| (I - G^*G)\beta \|_2 + \| G^*H\gamma \|_2 (8)
\]

Here, \( G \) and \( H \) are both i.i.d. matrices of Gaussian random variables (entries are linear projections of an isotropic Gaussian onto orthogonal vectors), each distributed according to \( N(0, \frac{1}{n}) \). \( G \) and \( H \) are also independent from each other. Hence, for the first term \( \| (I - G^*G)\beta \|_2 \) in (8), since \( n > d \), the columns of \( G \) are linearly independent with probability 1. So \( G^*G \) is full rank and

\[
\| (I - G^*G)\beta \|_2 = \| (I - (G^*G)^{-1}G^*G)\beta \|_2 = 0 \text{ with prob. 1 (9)}
\]

For the second term \( \| G^*H\gamma \|_2 \) in (8), we first note that

\[
\| G^*H\gamma \|_2 \leq \| G^* \|_op \| H\gamma \|_2 (10)
\]

where \( \| A \|_op = \max_{x\in\mathbb{R}^n, \|x\|_2=0} \{ \| Ax \|_\infty \} \). Hence, since \( G \) and \( H \) are independent,

\[
Pr(\| G^*H\gamma \|_2 \leq K_1K_2) \geq Pr(\| G^* \|_op \leq K_1) \cdot Pr(\| H\gamma \|_2 \leq K_2) (11)
\]

Since \( G \) is a \( n\times d \) matrix with \( d \leq n \) with entries i.i.d drawn from \( N(0, \frac{1}{n}) \), we use a theorem of Szarek [7], which states that the smallest singular value of \( G \), \( \sigma_{\text{min}}(G) \) obeys

\[
Pr\left( \sigma_{\text{min}}(G) > 1 - \sqrt{d/n - r} \right) \geq 1 - e^{-nr^2/2}
\]

so since we can verify that \( \| G^* \|_op = \| (G^*G)^{-1}G^* \|_op \leq \sigma_{\text{min}}(G) \), we have

\[
Pr\left( \| G^* \|_op \leq \frac{1}{1 - \sqrt{d/n - r}} \right) \geq 1 - e^{-nr^2/2} (12)
\]

Now consider \( \| H\gamma \|_2 \).

\[
\| H\gamma \|_2 = \left\| \begin{bmatrix} \langle e_1, v_{d+1} \rangle & \cdots & \langle e_1, v_q \rangle \\ \vdots & \ddots & \vdots \\ \langle e_n, v_{d+1} \rangle & \cdots & \langle e_n, v_q \rangle \end{bmatrix} \right\|_2 \\
= \frac{1}{n} \sum_{k=1}^{n} \sqrt{\sum_{i=1}^{q-d} \gamma_i v_{i+d}^2}
\]

Let \( a_k = (\sqrt{\gamma_i} v_{i+d}) \), so each \( a_k \) is a linear projection of a Gaussian random variable, and thus \( a_k \) is still Gaussian. Moreover, note \( E(a_k) = 0 \), \( \text{var}(a_k) = \gamma_i \), and \( \{a_k\}_{k=1}^n \) are independent. Each \( a_k \) is thus i.i.d. in \( N(0, \gamma_i) \).

Hence, \( E(\| H\gamma \|_2) = \| \gamma \|_2 \), \( \text{var}(\| H\gamma \|_2) = \frac{\| \gamma \|_2^2}{n} \). Using the one-sided Chebyshev inequality,

\[
Pr(\| W - E(W) \|_2 \geq b) \leq \frac{\text{var}(W)}{b^2} + b^2 (13)
\]

Combining (7), (8), (9), (11), (12), (14) proves the theorem. \( \square \)

While our measurement vectors may not always be i.i.d. Gaussian in feature space, we note that the above proof primarily relied on lower bounding \( \| G \|_op \) and upper bounding \( \| H \|_op \), which should also be possible for \( \Phi \) corresponding to many standard kernels. We hope to address this further in future work.

5. CONCLUSION

We have demonstrated that signals which show nonlinear sparsity, i.e. which can be represented in terms of a small number of components in feature space under an appropriate choice of kernel, can be reconstructed from far fewer random measurements than is typically required using traditional compressive sensing, indeed sometimes an order of magnitude fewer. Moreover, using the kernel trick framework, these gains can be achieved with very little computation, in a time comparable to that of traditional compressive sensing recovery algorithms. We hope to explore applications of this framework to other signals that show nonlinear sparsity, such as patches of natural images, in future work.

6. REFERENCES