This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

Compressed Signal Processing on Nyquist-sampled Signals

Jie Lu, *Student Member, IEEE*, Naveen Verma, *Member, IEEE*, and Niraj K. Jha, *Fellow, IEEE*

Abstract—Pattern-recognition algorithms from the domain of machine learning play a prominent role in embedded sensing systems, in order to derive inferences from sensor data. Very often, such systems face severe energy constraints. The focus of this work is to mitigate the computational energy by exploiting a form of compression which preserves a similarity metric widely used for pattern recognition. The form of compression is random projection, and the similarity metric is inner products between source vectors. Given the prominence of random projections within compressive sensing, previous research has explored this idea for application to compressively-sensed signals. In this work, we analyze the error sources faced by such approaches and show that the compressive-sensing setting itself introduces a significant source of feature-computation error (∼30%). We show that random projections can be exploited more generally without compressive sensing, enabling significant reduction in computational energy, and avoiding a significant source of error. The approach is referred to as *compressed signal processing (CSP)*, and it applies to Nyquist-sampled signals. We validate the CSP approach through two case studies. The first focuses on seizure detection using spectral-energy features extracted from electroencephalograms. We show that at a 32× compression ratio, the number of multiply-accumulate (MAC) and operand-access operations required is reduced by 21.2×, while achieving a sensitivity of 100%, latency of 4.33 sec, and false alarm rate of 0.22/hr; this compares to a baseline performance of 100%, 4.37 sec, and 0.12/hr, respectively. The second case study focuses on neural prosthesis based on extracting wavelet features from a set of detected spikes. We show that at a 32× compression ratio, the number of MAC and operand access computations required is reduced by 3.3×, while spike sorting performance can be maintained within an average error of 4.89% for spike count, 3.42% for coefficient of variance, and 4.90% for firing rate; this compares with a baseline average error of 4.00%, 2.75%, and 4.00% for spike count, coefficient of variance, and firing rate, respectively.

Index Terms—Classification, compressed signal processing, machine learning, Nyquist domain, random projections.

1 Introduction

MACHINE-learning algorithms make it possible to perform pattern recognition (e.g., classification) on data that is too complex to model analytically. Since such pattern recognition is of fundamental importance in diverse domains, these algorithms are having widespread impact. Our focus is on energy-constrained sensing applications, where the energy-intensive computations can quickly drain the battery. Taking the specific example of classification, Fig. 1(a) shows that two key steps are generally involved: (1) feature extraction and (2) classification [1]. Both steps can be energy-intensive, involving significant computation. Previous research has focused on lowering the energy consumption in these steps by compressing the input signals on which operations are applied. One such approach is to compress the incoming data and directly classify it, as shown in Fig. 1(b) [2]. The reduced dimensionality leads to fewer operations and thus lower system energy. However, this approach can only be applied to data that are directly classifiable and cannot be used for data that require feature extraction prior to classification. To overcome this, Shoaib et al. investigated direct signal processing on compressed data [3]. Taking the compressed data, the features are also extracted in compressed form and classified directly in that form, as shown in Fig. 1(c). For the applications targeted in that work, this reduces energy by one to two orders of magnitude while yielding classification performance comparable to traditional uncompressed analysis. However, the performance depends on the errors introduced as a result of the compressed form; a shortcoming of the approach in [3] is that significant (∼30%) error can be incurred in the similarity metric used for classification compared to uncompressed signal processing. Depending on the type of classifier and the data distributions of the applications, this could have a substantial impact on classification performance.

The type of compression used in work mentioned above [2], [3] is compressive sensing [4]. Use of random projections for signal analysis is a natural fit under the compressive-sensing setting. Random projections are relatively easy and energy-efficient to implement. What is notable for the approaches above is that they preserve the inner product between two vectors, within an error bound. The inner product between two vectors serves as a similarity metric, thereby providing a basis for direct classification. However, we present analysis that shows a dominant source of inner-product error in the method presented in [3] is the compressive-sensing formulation itself.

In this paper, we focus on reducing the signal processing-energy for feature-vector extraction, once again when classifiers that rely on the inner product between vectors are used. We show that random projections can be used to achieve this, but unlike the work in Shoaib et al., our focus is not

- *This work was supported by SRC, NSF (grant CCF-1253670), as well as AFOSR (grant FA9550-14-1-0293).*
- *This work was supported in part by Center for Future Architectures Research (C-FAR), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.*
- *Authors are with the Department of Electrical Engineering, Princeton University, Princeton, NJ, 08544 USA e-mail: {jielu, nverma, jha}@princeton.edu.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

on processing compressed data, but rather on processing uncompressed data. In addition to presenting a formulation that exploits random projection without compressive sensing, we show that a significant error affecting inner-product preservation is avoided as a result. We refer to this new approach as *compressed signal processing (CSP)*, and illustrate it in Fig. 1(d).

The objectives of this work are as follows:

1. Compression of linear transforms corresponding to linear signal-processing operations, thereby reducing the feature-vector dimensionality and thus the feature-extraction energy.
2. Ensuring minimal degradation in inner products between feature vectors (i.e., similarity metric for classification), and thereby ensuring accuracy of classification as the compression ratio is increased.

The paper is organized as follows. In Section II, we present background material on random projections and discuss prior works on direct compressed-domain analysis. In Section III, we provide a setup for CSP. In Section IV, we analyze the error sources within the previous approaches. In Section V, we describe the CSP approach. In Section VI, we evaluate our approach through two case studies. We conclude in Section VII.

2 Background

In this section, we provide background material on random projections and review related work on incorporating random projections in machine-learning algorithms for the purpose of classification.

2.1 Random Projection

Random projection has been extensively used in algorithm design, especially in approximation algorithms [5], [6], [7], [8]. It allows us to project the original N-dimensional data, $\boldsymbol{x}$, into a K-dimensional ($K \ll N$) subspace, using a random matrix $\boldsymbol{\Theta}$ of dimension $K \times N$:

$$\hat{\boldsymbol{x}} = \boldsymbol{\Theta}\boldsymbol{x} \tag{1}$$

The importance of random projection can be deduced from the Johnson-Lindenstrauss lemma [9]. This lemma states that if a set of points in an N-dimensional space are randomly projected onto a K-dimensional subspace, then the distances between the points, more specifically represented by the inner products, are preserved within an error bound [10]. An easy-to-implement projection uses random matrices drawn from a normal or Bernoulli distribution [4], [11]. As an example, from a Bernoulli distribution, a matrix can be constructed in the following way (Pr denotes probability):

$$\boldsymbol{\Theta} = \begin{cases} +1 & \text{with } Pr = \dfrac{1}{2} \\ -1 & \text{with } Pr = \dfrac{1}{2}. \end{cases} \tag{2}$$

Similarly, a matrix can be constructed from elements drawn from a normal distribution $\boldsymbol{\Theta} \sim N(0, 1)$.

2.2 Related Work

Previous works have exploited the inner-product preservation property of random projections within inference algorithms. This property is of significance because there exists a wide range of inference algorithms that rely on the inner product as a similarity metric. In linear classifiers, for example, the output decision of an input test vector is a scalar offset of an inner product between the weight vector and the input test vector. Many nonlinear classifiers utilize the inner product as the similarity metric as well. For example, pattern recognition algorithms that rely on kernel functions for nonlinear modeling of data [12] utilize the inner products of the data.

The K-nearest neighbor algorithm also relies on the Euclidean distance between vectors, which is a form of inner product. Kleinberg [7], for example, incorporates random projections into a nearest-neighbor classifier. The projections compress the training features extracted from a database of n points so that, given the features extracted from a query point, one can efficiently determine its nearest neighbors in the database.

Other researchers have also explored random projections in the context of data compression for direct classification [2], [6], [13], [14]. Unlike the approach in [7], these studies do not modify the classifiers, but rather focus on the data. However, since the compressed data is directly passed to the classifier, the intermediate feature extraction step is not considered. Thus, these approaches are only applicable to systems that do not require explicit feature extraction for high classification accuracy.

In many cases, feature extraction is required, and the signal processing this necessitates accounts for a significant fraction of the system energy. To extend the use of random projections to the whole system, Shoaib et al. [3] proposed a methodology whereby linear signal processing for feature extraction can be directly performed on the compressed data. Computational energy savings are obtained for both the feature extraction and classification stages. However, the downside of this approach is that the use of compressed data degrades the preservation of inner products. Across the various applications that may be of interest, dependence on the distribution of application data and the errors affecting the inner products can impact the generality of the approach.

3 Preliminaries

Fig. 1(a) shows a typical classification system, in which signal processing is carried out on a Nyquist-sampled signal. The input data are often segmented into epochs to perform epoch-by-epoch classification. Each epoch contains N time-domain samples, forming an N-dimensional vector. The appropriate features, with dimensionality M, are extracted from the N-dimensional vectors, and passed to a classification kernel. Feature extraction aims to create a representation that facilitates pattern recognition. This might correspond to dimensionality reduction (e.g., via principal component analysis) or emphasis (de-emphasis) of important (unimportant) variances (e.g., via projection to an alternate basis). Thus, feature extraction generally leads to better classification performance, often making it a crucial

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

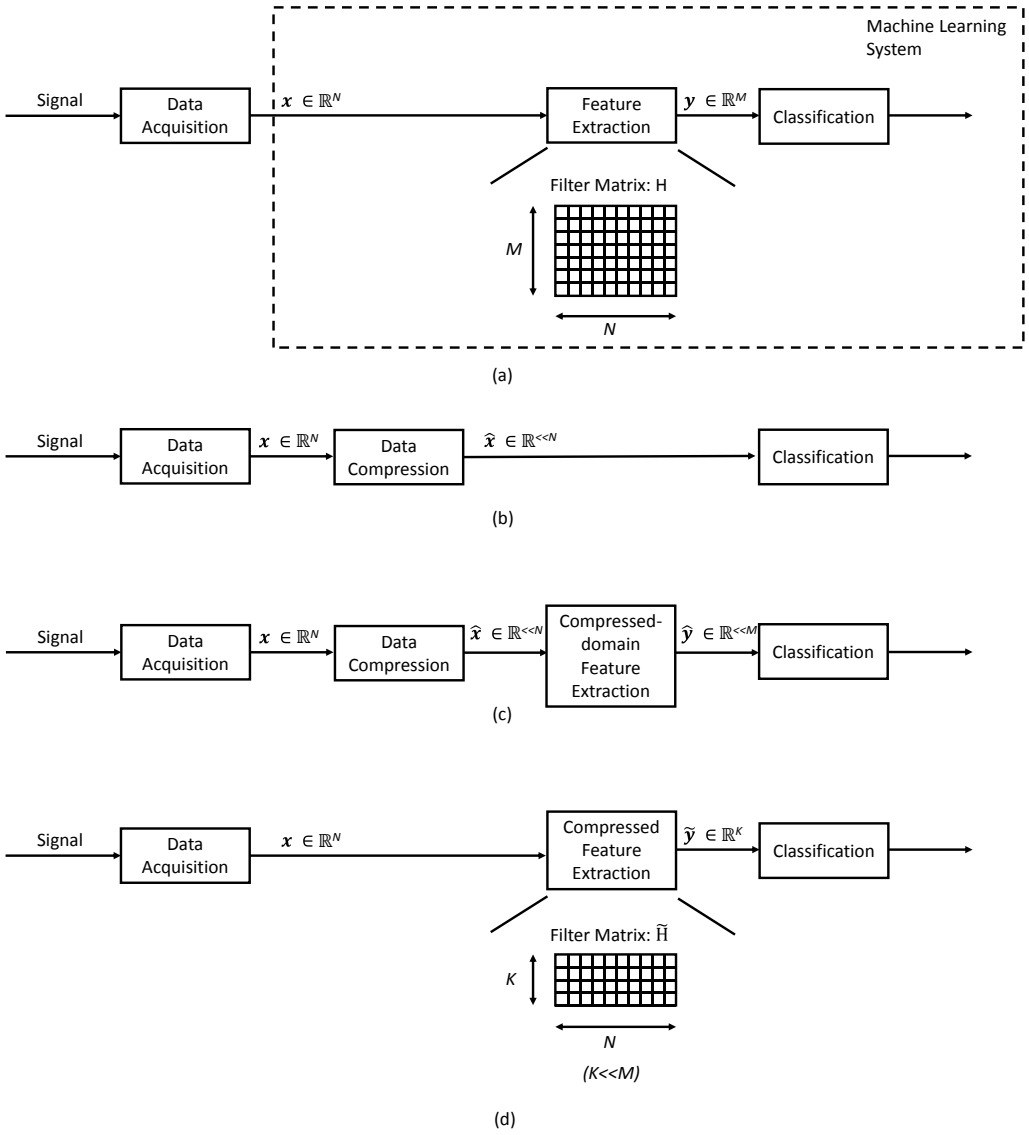

Fig. 1. Various classification systems: (a) traditional, (b) direct classification of compressed data (c) feature extraction in the compressively-sensed domain, and (d) compressed signal processing.

step in machine-learning systems. Our CSP approach primarily focuses on feature extraction based on linear signal processing, as required in a large class of applications. Our aim is to use the N-dimensional signals to compute a K-dimensional representation of the M-dimensional feature vector, where $K \ll M$, *but the information required for classification is retained* [Fig. 1(d)]. Lower dimensionality leads to fewer computations and thus reduces energy.

3.1 Feature Extraction

In order to achieve compression in the signal processing stage, we first need to consider how signal features are extracted. Generally, linear signal processing can be represented as a multiplication of an N-dimensional vector by a matrix. As an example, consider linear time-invariant (LTI) filtering. LTI filtering requires a convolution between the filter impulse response and the signal time samples. This requires roughly $J \times N$ multiply-accumulate (MAC) operations for a filter of order J and a signal of length N. Since the computational energy scales directly with the number of MAC operations, our aim is to reduce the latter, taken as a proxy for energy. Based on measurements from custom ICs that integrate accelerators, we have found that the total energy is well approximated by the number of MAC operations in applications [15], [16], [17]. For simplicity of analysis, we refer to both accessing of operands (possibly from a local memory) as well as the MAC computation itself as a MAC operation. While the specific energy depends on the technology used, we provide as an example values from a 130nm CMOS IC operating at 1.2V supply voltage, where one 32-bit MAC computation requires $11.8pJ$ and one operand access of 32kB memory requires $34.6pJ$ [15]. Thus, for one MAC computation and operand access, the amount of energy required is $46.4pJ$.

Circular convolution, when written as a series of MAC

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

operations, can be expressed by Eq. (3). For a given order-J filter and length-N signal ($J < N$),

$$y[j] = \sum_{i=0}^{N-1} h[(j-i)_N]x[i], \qquad j = \{0, 1, 2, \ldots, N-1\} \quad (3)$$

where h is the filter and x is the signal. Elements $h[j]$, where $J \leq j < N$, are zeros. This circular convolution ensures that the output is of size N, given two input sequences. Both input sequences have length N (or less) and sequence h is shifted modulo N [18]. Specifically, $h[(j-i)_N]$ means that for negative $j-i$ indices, each $h[j-i]$ is shifted to become the value at $h[N+(j-i)]$. Each $h[N+(j-i)]$ takes the value of the ith element of the original h filter, for $i, j = \{0, 1, 2, \ldots, N-1\}$. This is the shifted modulo N of circular convolution, as defined by Oppenheim and Schafer [18].

In order to exploit random projection in signal processing, we first represent convolution as matrix multiplication [3]. When processing an epoch of a signal, its Nyquist samples can be represented as a vector $\boldsymbol{x}$ of size N. The filter can be represented as an $M \times N$ matrix, $\boldsymbol{H}$, such that matrix multiplication between $\boldsymbol{H}$ and $\boldsymbol{x}$ is equivalent to the circular convolution operation. The matrix is a right circulant matrix where the rows represent shifted versions of the system impulse response. Then, the filtered output samples, represented by vector $\boldsymbol{y}$, are calculated through matrix multiplication:

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} \quad (4)$$

Thus formulated as a matrix transform, the filtering process can be compressed through random projection to decrease the dimensionality of the filtering matrix [Fig. 1(d)].

4 Error Analysis

As mentioned previously, Shoaib et al. [3] achieve low computational energy through direct processing of compressively-sensed signals. This approach compresses *both* dimensions of the signal-processing matrix $\boldsymbol{H}$. As we describe in the following section, our approach does not focus on processing compressively-sensed signals, and as a result compresses only one dimension of the signal-processing matrix. This might suggest that our proposed approach achieves less computational energy saving. In fact, however, compression of only one dimension results in a lower inner-product error. The reason for this is that compression in both dimensions involves use of a left pseudoinverse of the compressive random-projection matrix ($\boldsymbol{\Phi} \in \mathbb{R}^{L \times N}$), which is rank-deficient. Specifically, given an input vector $\boldsymbol{x}$ that is compressively sensed by random-projection matrix $\boldsymbol{\Phi} \in \mathbb{R}^{L \times N}$ ($L \ll N$), the $\boldsymbol{\Phi}$ matrix can be decomposed into $\boldsymbol{V}\boldsymbol{S}\boldsymbol{U}^{\mathsf{T}}$, where $\boldsymbol{V} \in \mathbb{R}^{L \times L}$, $\boldsymbol{S} \in \mathbb{R}^{L \times N}$, and $\boldsymbol{U} \in \mathbb{R}^{N \times N}$, using singular value decomposition. Then, by introducing a second random-projection matrix $\boldsymbol{\Theta} \in \mathbb{R}^{K \times M}$ ($K \ll M$), Shoaib et al. propose that the compressed signal-processing matrix $\hat{\boldsymbol{H}}$ be written as:

$$\hat{\boldsymbol{H}} = \boldsymbol{\Theta}\boldsymbol{H}\boldsymbol{U}\boldsymbol{S}^{-1}\boldsymbol{V}^{\mathsf{T}} \quad (5)$$

From Moore-Penrose pseudoinverse, we know that $\boldsymbol{U}\boldsymbol{S}^{-1}\boldsymbol{V}^{\mathsf{T}} = \boldsymbol{\Phi}^{\dagger}$. Thus we have the following:

$$\begin{aligned} &\langle \boldsymbol{H}\boldsymbol{x_1}, \boldsymbol{H}\boldsymbol{x_2} \rangle \\ \approx &\langle \hat{\boldsymbol{H}}\hat{\boldsymbol{x_1}}, \hat{\boldsymbol{H}}\hat{\boldsymbol{x_2}} \rangle \\ = &\langle \boldsymbol{\Theta}\boldsymbol{H}\boldsymbol{U}\boldsymbol{S}^{-1}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{\Phi}\boldsymbol{x_1}, \boldsymbol{\Theta}\boldsymbol{H}\boldsymbol{U}\boldsymbol{S}^{-1}\boldsymbol{V}^{\mathsf{T}}\boldsymbol{\Phi}\boldsymbol{x_2} \rangle \\ = &\langle \boldsymbol{\Theta}\boldsymbol{H}\boldsymbol{\Phi}^{\dagger}\boldsymbol{\Phi}\boldsymbol{x_1}, \boldsymbol{\Theta}\boldsymbol{H}\boldsymbol{\Phi}^{\dagger}\boldsymbol{\Phi}\boldsymbol{x_2} \rangle \end{aligned} \quad (6)$$

For the above approximation to hold, $\boldsymbol{\Phi}^{\dagger}\boldsymbol{\Phi}$ should be close to an identity matrix of size $N \times N$. However, this is generally limited by the rank deficiency of $\boldsymbol{\Phi}$. Namely, the Moore-Penrose pseudoinverse only ensures that $\boldsymbol{\Phi}^{\dagger}\boldsymbol{\Phi}$ preserves the characteristics that $\boldsymbol{\Phi}\boldsymbol{\Phi}^{\dagger}\boldsymbol{\Phi} = \boldsymbol{\Phi}$ and $\boldsymbol{\Phi}^{\dagger}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\dagger} = \boldsymbol{\Phi}^{\dagger}$ [19]; it does not generally ensure that $\boldsymbol{\Phi}^{\dagger}\boldsymbol{\Phi}$ equals the identity matrix.

In the context of classification operations, one should note that whereas Shoaib's approach may degrade the signal-to-noise ratio (SNR) of the inner-product estimate, classification performance may in fact be minimally affected. Zhuo et al. have demonstrated systems wherein errors in feature-vector estimation are well tolerated through appropriate training of the classifier, as long as the mutual information between feature vectors and their class membership is maintained [20]. Indeed, Shoaib et al. show that for their applications, mutual information is maintained up to substantial compression ratios, thus preserving classification performance [3]. Nonetheless, our approach avoids the above source of error, thus more generally addressing feature-vector extraction. We demonstrate the advantage of our approach by comparing its performance with that of Shoaib's in Section 6.

5 Compressed Signal Processing

When compressing the filtering matrix, our aim is to preserve the specific information required for classification, namely the inner product between vectors. As mentioned in Section 2, the inner product between vectors is preserved following random projection, within the error bound dictated by the Johnson-Lindenstrauss lemma. That is, for a feature vector $\boldsymbol{y}$ of dimensionality M, a random projection $\tilde{\boldsymbol{y}}$ of dimensionality K ($\ll M$) can be derived using a matrix $\boldsymbol{\Theta}$, whose elements may be drawn from a Bernoulli distribution of $\{-\frac{1}{\sqrt{K}}, \frac{1}{\sqrt{K}}\}$, or a normal distribution of $N(0, \frac{1}{K})$:

$$\tilde{\boldsymbol{y}} = \boldsymbol{\Theta}\boldsymbol{y}. \quad (7)$$

Then, the inner product between two vectors (from a set of n vectors), $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^M$, is estimated by the inner product between two vectors: $\boldsymbol{\Theta}\boldsymbol{u}, \boldsymbol{\Theta}\boldsymbol{v} \in \mathbb{R}^K$. According to the Johnson-Lindenstrauss lemma, for an $\epsilon \in (0, 1)$ and $K = O(\log n/\epsilon^2)$:

$$\begin{aligned} (1-\epsilon)\|\boldsymbol{u}-\boldsymbol{v}\|_2^2 &\leq \|\boldsymbol{\Theta}\boldsymbol{u}-\boldsymbol{\Theta}\boldsymbol{v}\|_2^2 \\ &\leq (1+\epsilon)\|\boldsymbol{u}-\boldsymbol{v}\|_2^2, \\ (1-\epsilon)\|\boldsymbol{u}+\boldsymbol{v}\|_2^2 &\leq \|\boldsymbol{\Theta}\boldsymbol{u}+\boldsymbol{\Theta}\boldsymbol{v}\|_2^2 \\ &\leq (1+\epsilon)\|\boldsymbol{u}+\boldsymbol{v}\|_2^2, \end{aligned} \quad (8)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

with probability $\geq 1 - 2e^{-(\epsilon^2-\epsilon^3)\frac{K}{4}}$ for each bound [21].

We thus have the error between the estimates and the actual inner products being bounded by the following:

$$|\langle \boldsymbol{\Theta u}, \boldsymbol{\Theta v}\rangle - \langle \boldsymbol{u}, \boldsymbol{v}\rangle| \geq \frac{1}{2}\epsilon(||\boldsymbol{u}||_2^2 + ||\boldsymbol{v}||_2^2)), \tag{9}$$

with probability $\leq 4e^{-(\epsilon^2-\epsilon^3)\frac{K}{4}}$. From Eq. (7) and the inner-product preservation property of the Johnson-Lindenstrauss lemma, we can say that for any two filtered signal vectors $\boldsymbol{y_1}, \boldsymbol{y_2} \in \mathbb{R}^M$:

$$\langle \boldsymbol{y_1}, \boldsymbol{y_2}\rangle \approx \langle \tilde{\boldsymbol{y_1}}, \tilde{\boldsymbol{y_2}}\rangle = \langle \boldsymbol{\Theta y_1}, \boldsymbol{\Theta y_2}\rangle, \tag{10}$$

and given Eq. (4) and Eq. (10),

$$\begin{aligned} \langle \boldsymbol{\Theta y_1}, \boldsymbol{\Theta y_2}\rangle &= \langle \boldsymbol{\Theta H x_1}, \boldsymbol{\Theta H x_2}\rangle \\ \boldsymbol{y_1}^\mathsf{T}\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{\Theta y_2} &= \boldsymbol{x_1}^\mathsf{T}\boldsymbol{H}^\mathsf{T}(\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{\Theta})\boldsymbol{H x_2} \\ &\approx \boldsymbol{x_1}^\mathsf{T}\boldsymbol{H}^\mathsf{T}\boldsymbol{H x_2} \\ &= \langle \boldsymbol{H x_1}, \boldsymbol{H x_2}\rangle \end{aligned} \tag{11}$$

The above approximation becomes equal if $\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{\Theta}$ is equal to the $M \times M$ identity matrix $\boldsymbol{I}$. Thus, we can solve for $\boldsymbol{\Theta}$ and construct the compressed filter matrix such that

$$\langle \boldsymbol{\Theta H}, \boldsymbol{\Theta H}\rangle = \langle \tilde{\boldsymbol{H}}, \tilde{\boldsymbol{H}}\rangle \approx \langle \boldsymbol{H}, \boldsymbol{H}\rangle \tag{12}$$

The compressed $\tilde{\boldsymbol{H}}$ contains fewer rows, as illustrated in Fig. 1(d). Thus, the filtering matrix $\boldsymbol{H}$ is compressed, implying the need for fewer computations. Specifically, instead of performing $M \times N$ MAC operations, we need to perform only $K \times N$ MAC operations ($K \ll M$), corresponding to a compression ratio of M/K.

Note in the formulation above, a random projection $\boldsymbol{\Theta}$, whose elements are drawn from a Bernoulli distribution, is employed. In general, other random projections can be chosen. As an example, simpler random projections [22], [23] may be considered; however, since the compressed signal-processing matrix $\tilde{\boldsymbol{H}}$ is computed once offline for a given system, the complexity of random projection is not critical. Ultimately, random projections that lead to low feature-vector estimation error are preferred.

5.1 Tradeoffs

CSP introduces specific trade-offs. First, as we decrease the dimension K, the overall signal processing energy decreases, but the inner-product error also increases. As expressed in the probability of the error bound, the inner-product error is impacted more heavily by the final compressed dimension K than the compression ratio M/K. Therefore, applications that involve a signal-processing matrix $\boldsymbol{H}$ with a large number of rows M can benefit more from this approach than applications whose signal-processing matrix has fewer rows to begin with. As an illustration, in Fig. 2, we examine the averaged SNR of the inner product estimate when compressing vectors using a random-projection matrix $\boldsymbol{\Theta}$ with dimension $M = 100$, 1000, and 10000. Vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^M$ are segments of electroencephalogram (EEG) signals that have been filtered by a Gaussian random matrix. The averaged SNR is defined as:

$$\text{SNR}_{\text{AVG}} = \frac{1}{n}\sum_{i,j=1}^{n} \frac{|\langle \boldsymbol{u_i}, \boldsymbol{v_j}\rangle|}{|\langle \boldsymbol{\Theta u_i}, \boldsymbol{\Theta v_j}\rangle - \langle \boldsymbol{u_i}, \boldsymbol{v_j}\rangle|} \tag{13}$$

where n is the number of vectors over which the average is taken.

As the size of compressed matrix increases (i.e. increasing dimension K), the average SNR increases linearly with respect to the square root of K (Fig. 2). This can be understood as follows. The error in the inner product estimate [i.e., denominator of Eq. (13)] is due to $\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{\Theta}$ deviation from the identity matrix, according to Eq. (11). Given that, in our illustration, elements from $\boldsymbol{\Theta}$ of dimension $K \times M$ are sampled from a Bernoulli distribution of $\{-\frac{1}{\sqrt{K}}, \frac{1}{\sqrt{K}}\}$ with equal probabilities, we know that $\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{\Theta}$ has diagonal elements equal to 1. Thus, the elements that contribute towards the error are the off-diagonal elements of $\boldsymbol{\Theta}^\mathsf{T}\boldsymbol{\Theta}$, which have a mean of zero and a standard deviation of $\frac{1}{\sqrt{K}}$. This scaling of the standard deviation accounts for the observed SNR behavior.

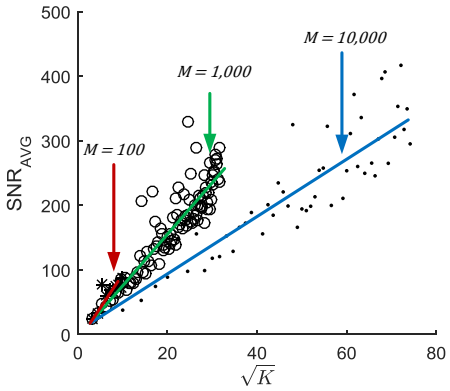

Fig. 2. Averaged SNR vs. $\sqrt{K}$ for randomly chosen filtering matrices. The $\boldsymbol{H}$ dimensions varied in the range M = 100, 1000 and 10000. The lines demonstrate the linear relationship between $\sqrt{K}$ and averaged SNR for different $\boldsymbol{H}$ dimensions.

Further, the dependence of the SNRs shown in Fig. 2 for different M can be understood as follows. First, higher values of M increase the likelihood of two vectors being orthogonal. This leads to near-zero values for the numerator of Eq. (13), causing lower overall SNR. Second, the SNR is also dependent on the error term in the denominator of Eq. (13). Specifically, we can see from Eq. (9), that this error term, increases with the Euclidean norm of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. Since the Euclidean norm increases with M, the SNR decreases correspondingly. If we were to normalize vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ such that $||\boldsymbol{u}||_2^2 = 1$ and $||\boldsymbol{v}||_2^2 = 1$, then the averaged error over different $\boldsymbol{u}$ and $\boldsymbol{v}$ vectors [denominator of Eq. (13)] still shows a linear relationship with $\frac{1}{\sqrt{K}}$, but it would no longer vary with different values of M.

5.2 Feature Selection

Before moving on to discuss classification of the features derived from CSP, we briefly discuss a common feature-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

reduction methodology – feature selection (FS). It refers to the selection of a subset of the original features that satisfies a predefined evaluation criterion. The evaluation criterion often depends on the relevant information within the feature set (e.g., entropy, performance accuracy, Fisher score, etc.). By selecting only a subset, the calculations for non-selected features can be ignored, thus reducing the number of calculations needed for feature extraction. Since our approach focuses on compression of the feature extraction step, it is also applicable in the context of FS. In this case, we would only apply CSP to extraction of the selected features. We evaluate this hybrid approach in Section 6.

5.3 Classification

Though the number of MAC operations can be substantially reduced for feature extraction, depending on the number of MAC operations required for classification, it is possible that the overall savings could be marginal. Thus, to evaluate the full system, our study employs applications using two practical classification algorithms. The first is an ensemble classifier based on the adaptive boosting (AdaBoost) algorithm, which utilizes a set of weighted weak classifiers. A weak classifier is defined as a classifier that incurs substantial error, typically due to an inadequate ability to fit to training data [24]. AdaBoost can be combined with any classifier and, in practice, has been used with classifiers such as neural nets and decision trees. For this paper, we have used decision trees with seven or fewer nodes as the weak classifier. The decision tree is chosen both because it has empirically been shown to perform well with AdaBoost [25] and because it requires only a modest computation to carry out classification. Specifically, each node of the decision tree requires one comparison operation, whose energy includes the energy of the comparison and the energy to access the operands used for the comparison. A comparison operation is much simpler than a MAC operation. In addition to the comparisons required within the weak classifiers, each iteration of boosting requires an additional MAC operation to apply the weight of each weak classifier and then perform voting via addition over the weak-classifier decisions. The second classifier is K-means, which derives cluster centroids during the training phase. These centroids are used for cluster assignment during the testing phase. Hence, the number of MAC operations scales with the number of clusters and the feature-vector size. In the case studies considered ahead, we show that the computations required for classification are far fewer than feature extraction. Thus, feature extraction is indeed the limiting computation. Hence, substantial system-level savings can be achieved through the proposed CSP approach.

6 Case Studies

To evaluate our methodology, we pursue two case studies. The first is seizure detection based on scalp EEG recording, and the second is spike sorting based on action-potential recordings.

The signal processing matrix is compressed by obtaining $\tilde{\boldsymbol{H}}$ through Eq. (12). Elements of the projection matrix $\boldsymbol{\Theta}$ are drawn based on Eq. (2).

6.1 Case 1: EEG-based Seizure Detection

We obtained EEG signals for three patients from the CHB-MIT database for the purpose of seizure detection [26]. For each patient, we used 18 channels of EEG signals. For each channel, we segmented continuous EEG signals into three-second signal epochs with two-second overlap between consecutive epochs. Each second of EEG data contains 256 time-domain samples, resulting in 768 samples per epoch. We processed the samples through eight band-pass filters (BPFs), each with a passband width of 3 Hz, and a center frequency in the 0-24 Hz range. However, prior to bandpass filtering, we need to first decimate the EEG data to 64 Hz. Decimation is required to reduce the computational requirement of the eight BPFs. Once the decimated signal is passed through one of the BPFs, only 128 coefficients of the filtered signal are used in order to minimize the edge effect. The features correspond to the energy of the output samples from each filter over the epoch. This requires summing the squared values of the filter output samples. Note that this does not correspond to linear signal processing to which our CSP approach directly applies. In fact, the seizure-detection system represents a special case, wherein feature extraction involves two distinct stages: (1) linear filtering, to which CSP applies, and energy savings can thus be obtained; and (2) energy accumulation, which can be computed by treating the filter output samples as a vector whose inner product is taken with *itself*. Thus, inner-product preservation is exploited to compute the features themselves rather than to compute the classification result. Classification then proceeds as usual on the resulting features.

Specifically, the extracted features are provided to an AdaBoost classifier based on decision-tree weak classifiers. For each epoch of data, 768 signal samples per EEG channel are processed to produce eight features (one from each BPF). Thus, a total of 144 features are extracted for every three-second epoch from data over 18 EEG channels. We evaluate detection performance of the system through the following three metrics: sensitivity, latency, and number of false alarms per hour. Sensitivity is defined as the percentage of correctly detected seizures over the entire seizure-detection period. Latency is defined as the number of seconds elapsed between the expert-identified seizure onset and the system-detected seizure onset. Number of false alarms per hour is defined as the number of falsely detected seizures averaged over the total number of hours within the EEG recording.

6.1.1 Matrix Form for Feature Extraction

To implement compression in the signal processing stage, we first represent the LPF required for decimation in matrix form. The LPF is of order 256. We then represent the required eight BPFs in matrix form. Each BPF is of order 64. To represent the LPF and each BPF computation as a matrix operation, we generate the LPF matrix, $\boldsymbol{H_{LPF}}$, and BPF matrix, $\boldsymbol{H_{BPF}}$, using Eq. (3). The first row of each filter matrix is based on the original filter and each subsequent row is a right-shifted version of the previous row, thus forming a right circulant matrix. However, instead of using every row of $\boldsymbol{H_{LPF}}$, we only take every fourth row. Multiplication of this reduced matrix with the signal is equivalent to decimation of the signal by a factor of 4. We

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

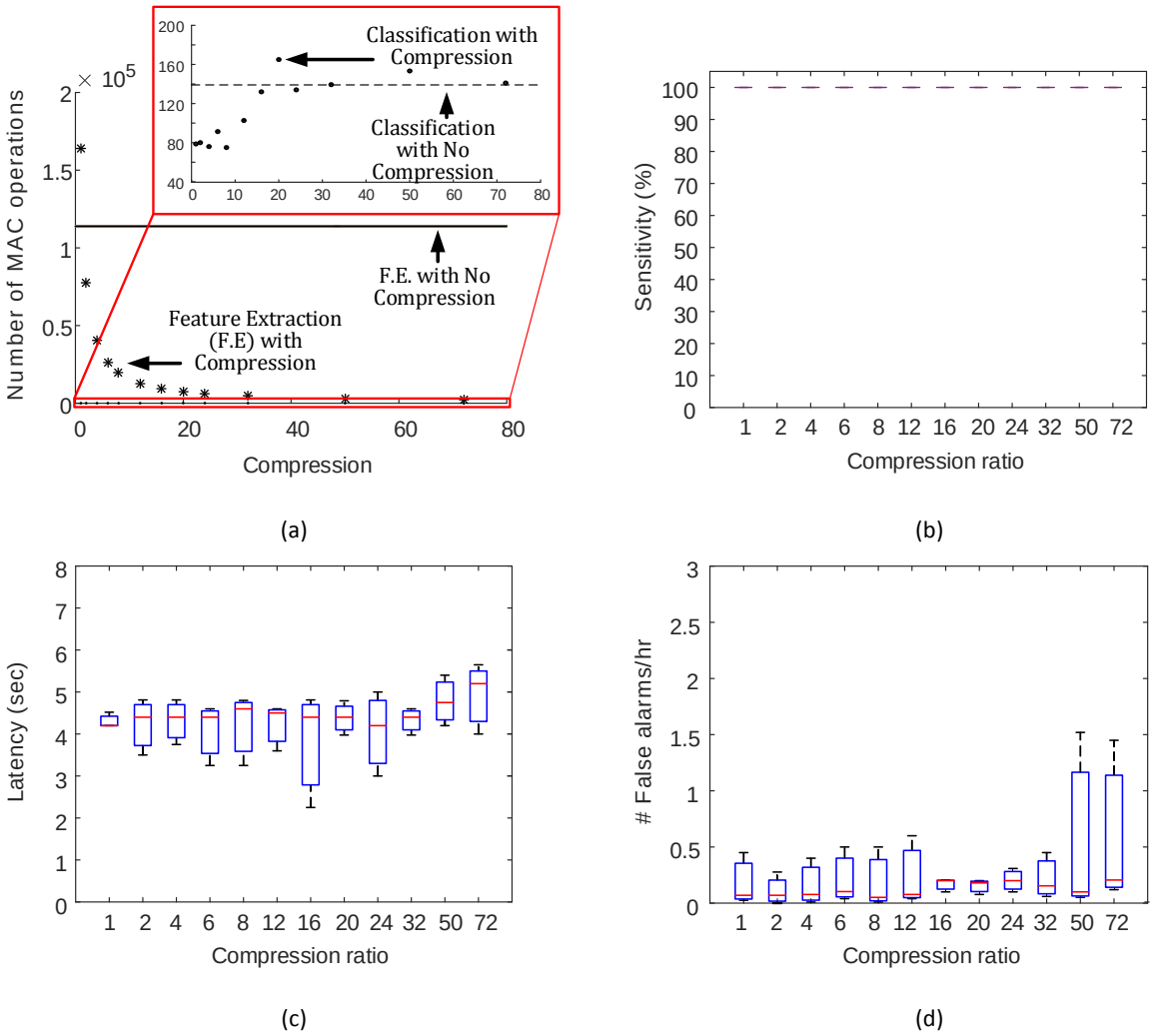

Fig. 3. Seizure-detection performance with respect to compression ratio using an AdaBoost classifier based on decision trees. (a) Feature extraction and classification energy reduction represented by the reduction in the number of MAC operations achieved using the $\tilde{\boldsymbol{H}}$ matrix with increasing compression ratios. Solid and dashed lines represent the number of MAC operations required by uncompressed feature extraction and classification methods, respectively. (b) Sensitivity vs. compression ratio. (c) Detection latency vs. compression ratio. (d) Number of false alarms per hour vs. compression ratio.

denote this reduced matrix as $\boldsymbol{H_D}$. Eight final filter matrices can be produced by multiplying each BPF with the LPF (i.e. $\boldsymbol{H} = \boldsymbol{H_{BPF}}\boldsymbol{H_D}$).

6.1.2 Results

Potential system energy savings are investigated first. System energy can be decomposed into feature-extraction energy and classification energy. As mentioned before, the MAC operation serves as a proxy for the energy required. The number of MAC operations required for feature extraction is dependent on the size of the eight compressed filters. The number of MAC operations decreases substantially with increasing compression ratios because not only is the number of multiplications in the filtering matrix reduced, but so is the number of multiplications required to compute the inner product for energy accumulation [Fig. 3(a)]. Classification energy, on the other hand, depends on the number of iterations required for the AdaBoost classifier to achieve convergence. For the decision tree used, the maximal amount of energy per decision tree (seven nodes) is that of seven comparisons. Both because this number is small and because comparison operations consume much lower energy relative to multiplications, we have only taken into consideration the number of MAC operations required per iteration for weighted voting over the weak classifiers. The AdaBoost classifier with no compression of features requires an average of 138 iterations (i.e., 138 MAC operations). With an increase in the compression ratio, the number of MAC operations per classification does not change substantially, increasing to just 141 at a 72× compression ratio [Fig. 3(a)]. Overall, at a 32× compression ratio, the total number of MAC operations is reduced by 21.2×. This reduction is defined as the ratio of the number of MAC computations required to process and classify the signals without any compression, to the number of MAC computations required to compressively process and classify the signals with CSP.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

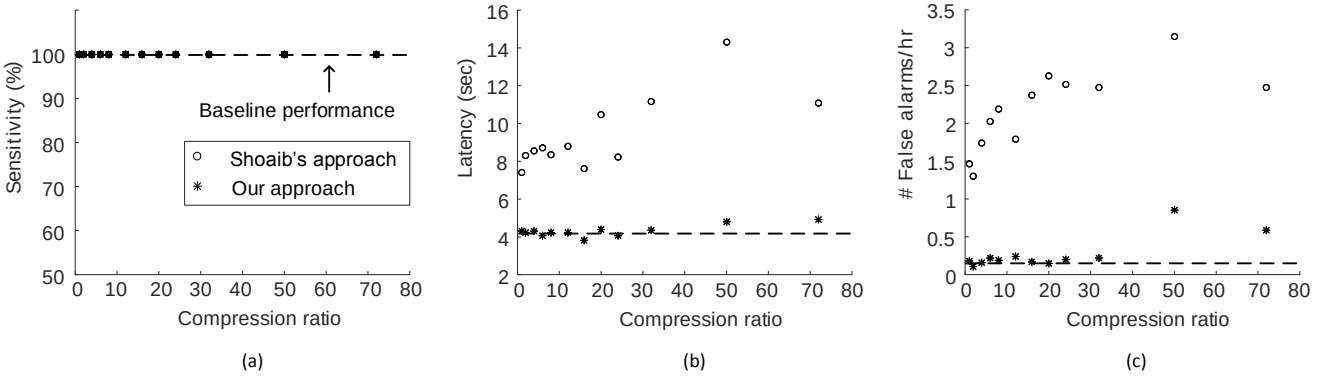

Fig. 4. Comparison between previous approach and our approach for seizure-detection performance with respect to compression ratio. Dotted line indicates baseline performance where no compression is used. (a) Sensitivity vs. compression ratio. (b) Detection latency vs. compression ratio. (c) Number of false alarms per hour vs. compression ratio.

Performance is evaluated using three criteria: sensitivity, latency, and number of false alarms per hour [Fig. 3(b), (c), (d)]. Uncompressed signal processing matrices are used to establish a baseline (sensitivity = 100%, latency = 4.37s, and false alarms per hour = 0.12). As an example, at a 32× compression ratio, a similar performance is achieved: 100% sensitivity, 4.33s latency, and 0.22 false alarms per hour.

6.1.3 Comparison to Previous Approaches

Next, we show how our system performs in comparison with the work proposed by Shoaib et al. [3]. Fig. 4 compares the sensitivity, latency, and false alarm per hour of the two approaches as the compression ratio is increased. For example, at 32× compression ratio, Shoaib's approach achieves an average performance of 100% sensitivity, 11.14s latency, and 2.48 false alarms per hour. Thus, we see that both the latency and false alarms per hour degrade significantly compared to our approach. Note that the two approaches are not interchangeable. Specifically, we cannot simply replace the feature extraction step of Shoaib's approach with that of our approach since Shoaib's approach is targeted to compressed input signals whereas ours is targeted at Nyquist-domain signals.

In this application, FS can be performed regardless of CSP, since CSP is simply applied to compute all features selected (i.e., the umber of features is not changed). However, in applications where our approach impacts the feature-vector dimensionality, one could employ FS within $\boldsymbol{H}$ prior to CSP. This means that further dimension reduction can be achieved after feature selection is done. We demonstrate this with the next case study.

6.2 Case 2: System for Neural Prosthesis

For the second case study, we investigated a neural prosthesis application that is based on spike detection and sorting. A typical system is composed of four computational blocks: spike detection/alignment, sorting, analysis, and prosthesis control. It is preferable to perform spike detection/alignment on an implant to address transmission energy and bandwidth constraints in most modern systems [27]. Once aligned, spikes can be sorted on an external head

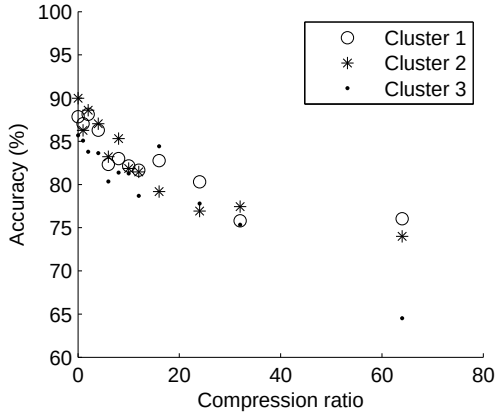

Fig. 5. Cluster accuracy achieved by classification of compressively extracted features with respect to increasing compression ratio.

stage or within the implant prior to analysis. Spike sorting is done using discrete wavelet transform (DWT) and clustered using the K-means algorithm [28]. Using these methods, the spikes are sorted into different groups where each group contains all the spikes produced by an individual neuron. Once the spikes are sorted, feature parameters can be extracted. These parameters include spike count (SC), neuron firing rate (FR), and coefficient of variation (CV), which can be used to drive prosthesis control [29]. We refer to these parameters as control parameters. For this case study, we focus on MAC operation reduction during the spike sorting procedure. We use the control parameters to demonstrate performance. We consider a total of 18 cases with differing recognition difficulties and noise levels.

6.2.1 The DWT Matrix

The DWT of a signal is often implemented using a filter bank. The filter bank is derived from a four-level decomposition of a Haar mother wavelet. The

four-level decomposition can be obtained through a series of low-pass filters (LPFs) and high-pass filters (HPFs). The signal is first passed through an HPF and down-sampled by 2×, filtering out the lower half of the frequency

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

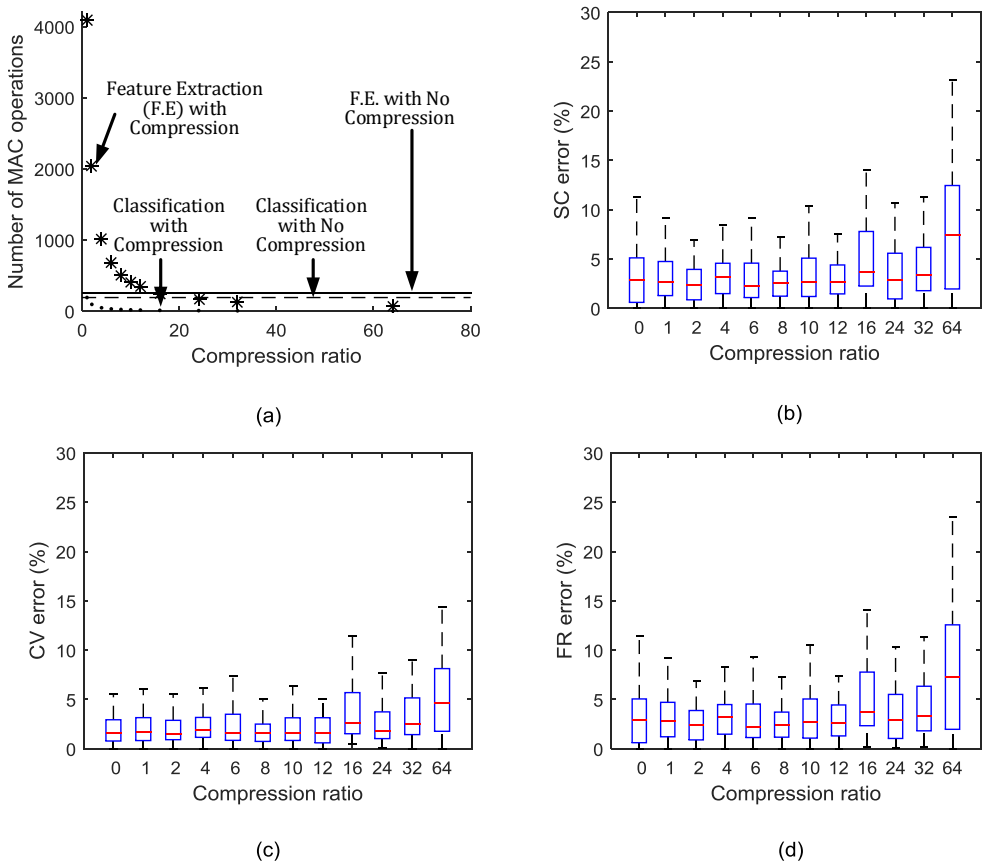

Fig. 6. Three-cluster K-means performance with an increasing compression ratio. (a) Signal processing and classification energy reduction represented by the number of MAC operations achieved using the $\tilde{H}$ matrix with increasing compression ratios. Solid and dashed lines represent the number of MAC operations required by uncompressed feature extraction and classification methods, respectively. (b) Spike count percentage error compared to ground truth vs. compression ratio. (c) Coefficient of variance percentage error compared to ground truth vs. compression ratio. (d) Firing rate percentage error compared to ground truth vs. compression ratio.

bank. This comprises one level of wavelet decomposition. The same process is repeated on the LPF outputs to achieve further decomposition [30]. In order to formulate the entire process as a matrix operation on a vector derived from Nyquist sampling of a signal, we once again map convolution to a right-circulant matrix. Convolution, either linear or circular, can be realized in matrix form. For DWT, the first half of the matrix ($M/2 \times N$) can represent the HPF with every other right-circulant row retained, corresponding to output down-sampling. The remaining matrix then similarly represents HPFs and LPFs, appropriately down-sampled. This way, a DWT matrix can be formed, as a cascade of HPFs and LPFs, all appropriately down-sampled [3].

6.2.2 Results

The spike sorting system is implemented for records E1, E2, D1, and D2, from simulated neurological data [28]. All signals are first processed for detection and alignment of the spikes using a thresholding algorithm, as described by Quiroga et al. [28]. This processing produces 64 data samples per detected spike. We compare experimental results for uncompressed signal processing and CSP on the spike data. All spikes from each record are clustered into three clusters. The accuracy of each cluster, defined as the percentage of correctly clustered spikes compared to the ground truth, is analyzed for various compression ratios. With no compression, the spike-sorting system correctly clusters, on an average, 87.84% of the neurons. With increasing compression ratio, the clustering accuracies decrease to 81.72%, 76.16%, and 71.51% at $10\times$, $32\times$, and $64\times$ compression ratios, respectively (Fig. 5).

Next, we analyze the control parameters (SC, CV, and FR) over all records for each compression ratio (Fig. 6). We first consider the system energy savings with increasing compression ratios [Fig. 6(a)]. The number of MAC operations decreases with decreasing feature-extraction matrix size ($K \times N$). Thus, as the compression ratio is increased, the MAC-operation ratio decreases. In this system, the classification energy is also affected by the reducing feature-extraction matrix size. As mentioned earlier, an increase in the compression ratio implies a smaller feature-vector size, and the energy of K-means classification scales with the feature-vector size (in addition to the number of cluster

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

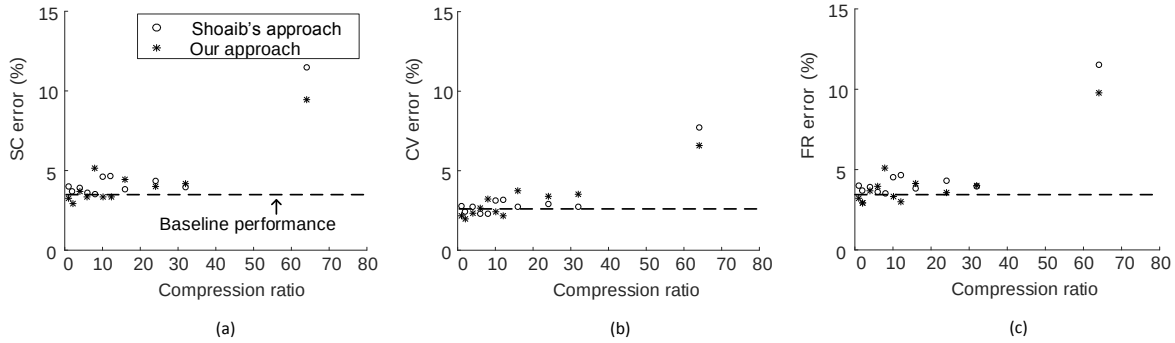

Fig. 7. Three-cluster K-means performance comparison between Shoaib's approach and our approach with an increasing compression ratio. The dotted line represents baseline performance when no compression is implemented. (a) Spike count percentage error compared to ground truth vs. compression ratio. (b) Coefficient of variance percentage error compared to ground truth vs. compression ratio. (c) Firing rate percentage error compared to ground truth vs. compression ratio.

centroids). Therefore, classification energy is also reduced with an increase in the compression ratio. Overall, at a 32× compression ratio, the total number of MAC operations is reduced by 3.3×. The number of MAC operations is not as aggressively reduced as the compression ratio due to the sparsity of the DWT matrix. For a more dense filtering matrix, we would observe computational savings comparable to the compression ratio.

To evaluate the performance, we first compute the control parameters for ground-truth clusters, which we call ground-truth control parameters. The extracted control parameters for all compression ratios are compared to the ground-truth control parameters. Since the clustering algorithm is not perfect, there are errors in the control parameters even in the case of no compression. With no compression, the extracted control parameters correspond to overall errors of 4.00% for SC, 2.75% for CV, and 4.00% for FR, compared to ground-truth control parameters. As shown in Fig. 6, the overall control parameter errors are within 5% for all counts at a 32× compression ratio, and within 10% at a 64× compression ratio.

6.2.3 Comparison to Previous Approaches

Next, we evaluate CSP relative to the approach presented by Shoaib et al. [3]. Fig. 7 shows the comparisons. Both approached achieve near-baseline performance for up to 32× compression ratio. Even at 64× compression ratio, both approaches maintain average FR, SC, and CV errors under 15%. Thus, for this case study, the two approaches are comparable.

Next, we demonstrate how FS can be used in conjunction with CSP. For FS, we used Lilliefors modification of the Kolmogorov-Smirnov (KS) test for normality, as proposed by Quiroga et al. [28]. The test compares the cumulative distribution function of the data with that of a Gaussian distribution with the same mean and variance. Coefficients with the largest deviation from normality are used. This test is employed because coefficients should have a multimodal distribution and thus ones that are furthest from a normal distribution would best separate the different spike classes. Using the above-mentioned feature selection method, we compare CSP, FS, and a hybrid approach (FS + CSP). For FS and the hybrid approach, we have two levels of FS in terms of the number of feature selected. For one level, we select half of the features (32 features) out of the original feature vector (i.e., our feature-extraction matrix after FS has only half the feature dimension relative to the original feature-extraction matrix). For the second level, we select a third of the features (21 features) out of the original feature vector. For the hybrid approach, we apply our approach on top of the two levels of FS and analyze the performance over different number of features. The number of features decreases as we increase the compression ratio. Fig. 8 shows the spike sorting performance using the hybrid approach compared to CSP with no FS. We see that the performance of CSP alone and the hybrid approaches for 32 features is comparable. However, FS of 21 features produces much higher error than either CSP alone or the hybrid approach. Interestingly, CSP seems to correct, to a certain degree, the error caused by FS.

7 Conclusions

In this paper, we presented the CSP approach in which random projections are used to significantly reduce the amount of computations involved in feature extraction, while incurring minimal performance loss. We demonstrated that CSP is able to overcome feature extraction errors that are inherent in a previous approach employing random projections. This leads to significant reductions in overall system energy. We evaluated the proposed methodology through two case studies. Both studies demonstrated acceptable performance even at 32× compression ratios. The CSP approach focuses on the feature-extraction stage of the classification system. The energy saving (if any) in the classification stage depends on the type of classifier used. We also demonstrate the compatibility between FS and CSP.

References

[1] S. K. Mitra, *Digital Signal Processing, Fourth Edition*. McGraw Hill, 2010.

[2] R. Calderbank, S. Jafarpour, and R. Schapire, "Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain," *Technical Report, Princeton University*, Dec. 2009.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2016.2532861, IEEE Transactions on Computers

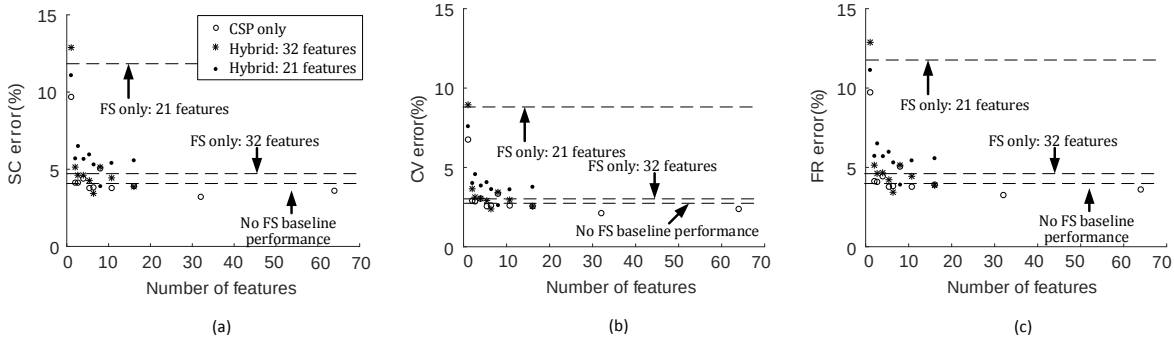

Fig. 8. Three-cluster K-means performance with an increasing compression ratio. We demonstrate three cases: no FS, FS with 32 features, FS with 21 features, all shown in dashed lines. CSP is then applied to all three cases for further dimension reduction. (a) Average spike count percentage error compared to ground truth vs. number of features. (b) Average coefficient of variance percentage error compared to ground truth vs. number of features. (c) Average firing rate percentage error compared to ground truth vs. number of features.

[3] M. Shoaib, N. K. Jha, and N. Verma, "Signal processing with direct computations on compressively sensed data," *IEEE Trans. VLSI Systems*, vol. 23, no. 1, pp. 30–43, Jan. 2015.

[4] E. J. Candès, "Compressive sampling," in *Proc. Int. Congress of Mathematicians*, Aug. 2006, pp. 1433–1452.

[5] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," in *Proc. 30th ACM Symp. Theory of Computing*, Jan. 1998, pp. 604–613.

[6] D. Fradkin and D. Madigan, "Experiments with random projections for machine learning," in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Aug. 2003, pp. 517–522.

[7] J. M. Kleinberg, "Two algorithms for nearest-neighbor search in high dimensions," in *Proc. 29th Annual ACM Symp. Theory of Computing*, May 1997, pp. 599–608.

[8] T. Sarlos, "Improved approximation algorithms for large matrices via random projections," in *Proc. 47th IEEE Symp. Foundations of Computer Science*, Oct. 2006, pp. 143–152.

[9] W. Johnson, J. Lindenstrauss, and G. Schechtman, "Extensions of Lipschitz maps into Banach spaces," *Israel J. Mathematics*, vol. 54, no. 2, pp. 129–138, Jun. 1986.

[10] S. Dasgupta and A. Gupta, "An elementary proof of a theorem of Johnson and Lindenstrauss," *Random Structures & Algorithms*, vol. 22, no. 1, pp. 60–65, Jan. 2003.

[11] D. Achlioptas, "Database-friendly random projections," in *Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, May 2001, pp. 274–281.

[12] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, Jun. 1999.

[13] S. Deegalla and H. Bostrom, "Reducing high-dimensional data by principal component analysis vs. random projection for nearest neighbor classification," in *Proc. 5th Int. Conf. Machine Learning and Applications*, Dec. 2006, pp. 245–250.

[14] X. Z. Fern and C. E. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proc. 20th Int. Conf. Machine Learning*, Aug. 2003, pp. 186–193.

[15] K. H. Lee and N. Verma, "A low-power processor with configurable embedded machine-learning accelerators for high-order and adaptive analysis of medical-sensor signals," *IEEE J. Solid-State Circuits*, vol. 48, no. 7, pp. 1625–1637, July 2013.

[16] M. Shoaib, K. H. Lee, N. Jha, and N. Verma, "A 0.6 - 107 μW energy-scalable processor for directly analyzing compressively-sensed eeg," *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 61, no. 4, pp. 1105–1118, Apr. 2014.

[17] K. H. Lee and N. Verma, "A low-power microprocessor for data-driven analysis of analytically-intractable physiological signals in advanced medical sensors," in *Proc. Symp. VLSI Circuits*, June 2013, pp. C250–C251.

[18] A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing, Second Edition*. Prentice Hall, 1989.

[19] J. Barata and M. Hussein, "The Moore-Penrose pseudoinverse: A tutorial review of the theory," *Brazilian J. Physics*, vol. 42, no. 1-2, pp. 146–165, 2012.

[20] Z. Wang, K. H. Lee, and N. Verma, "Overcoming computational errors in sensing platforms through embedded machine-learning kernels," *IEEE Trans. VLSI Systems*, vol. 23, no. 8, pp. 1459–1470, Aug. 2014.

[21] M. Balcan, A. Blum, and S. Vempala, "Kernels as features: On kernels, margins, and low-dimensional mappings," in *Proc. 15th Int. Conf. Algorithmic Learning Theory*, Oct. 2004, pp. 194–205.

[22] N. Ailon and B. Chazelle, "Faster dimension reduction," *Commun. ACM*, vol. 53, no. 2, pp. 97–104, Feb. 2010.

[23] J. A. Tropp, "Improved analysis of the subsampled randomized Hadamard transform," *Advances in Adaptive Data Analysis*, vol. 3, no. 01–02, pp. 115–126, Jul. 2011.

[24] R. E. Schapire and Y. Freund, *Boosting, Foundations and Algorithms*. Massachusetts Institute of Technology Press, 2012.

[25] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. 23rd Int. Conf. Machine Learning*, Jun. 2006, pp. 161–168.

[26] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, Physiotoolkit, and Physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[27] R. R. Harrison, P. T. Watkins, R. J. Kier, D. J. Lovejoy, D. J. Black, B. Greger, and F. Solzabacher, "A low-power integrated circuit for a wireless 100-electrode neural recording system," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 123–133, Jan. 2007.

[28] R. Q. Quiroga, Z. Nadasdy, and Y. Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–1687, Aug. 2004.

[29] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, no. 5681, pp. 258–262, Jul. 2004.

[30] T. K. Sarkar, C. Su, R. S. Adve, M. Salazar-Palma, L. E. Garcia-Castillo, and R. R. Boix, "A tutorial on wavelets from an electrical engineering perspective I: Discrete wavelet techniques," *IEEE Mag. Antennas and Propagation*, vol. 40, no. 5, pp. 49–68, Oct. 1998.

Jie Lu received her B.Sc. (Hons.) degree in life sciences from Queens University, Kingston, Canada, in 2010, and the M.A.Sc. degree in biomedical engineering from the University of Toronto, Toronto, Canada, in 2013. She is currently a Ph.D. student in electrical engineering at Princeton University, Princeton, USA. Her main area of research interest cover the implementation of energy-efficient platforms for sensing and computing. She has presented at 2013 International BCI meeting and published in Frontiers on brain-computer interfaces. During her masters and undergraduate degrees, she has received James F. Crothers Fellowships in Perpipheral Nerve damage 2012, Kimel Family Graduate Student Scholarship 2012, and NSERC-USRA 2008.

Naveen Verma received the B.A.Sc. degree in Electrical and Computer Engineering from the University of British Columbia, Vancouver, Canada in 2003, and the M.S. and Ph.D. degrees in Electrical Engineering from Massachusetts Institute of Technology in 2005 and 2009 respectively. Since July 2009 he has been with the department of Electrical Engineering at Princeton University, where he is currently an Associate Professor. His research focuses on advanced sensing systems, including low-voltage digital logic and SRAMs, low-noise analog instrumentation and data-conversion, large-area sensing systems based on flexible electronics, and low-energy algorithms for embedded inference, especially for medical applications. Prof. Verma is recipient or co-recipient of the 2006 DAC/ISSCC Student Design Contest Award, 2008 ISSCC Jack Kilby Paper Award, 2012 Alfred Rheinstein Junior Faculty Award, 2013 NSF CAREER Award, 2013 Intel Early Career Award, 2013 Walter C. Johnson Prize for Teaching Excellence, 2013 VLSI Symp. Best Student Paper Award, and 2014 AFOSR Young Investigator Award.

Niraj K. Jha (S'85-M'85-SM'93-F'98) received his B.Tech. degree in Electronics and Electrical Communication Engineering from Indian Institute of Technology, Kharagpur, India in 1981, M.S. degree in Electrical Engineering from S.U.N.Y. at Stony Brook, NY in 1982, and Ph.D. degree in Electrical Engineering from University of Illinois at Urbana-Champaign, IL in 1985. He is a Professor of Electrical Engineering at Princeton University. He is a Fellow of IEEE and ACM. He has served as the Editor-in-Chief of IEEE Transactions on VLSI Systems and an Associate Editor of IEEE Transactions on Circuits and Systems I and II, IEEE Transactions on VLSI Systems, IEEE Transactions on Computer-Aided Design, and Journal of Electronic Testing: Theory and Applications. He is currently serving as an Associate Editor of IEEE Transactions on Computers, Journal of Low Power Electronics, and Journal of Nanotechnology. He has also served as the Program Chairman of the 1992 Workshop on Fault-Tolerant Parallel and Distributed Systems, the 2004 International Conference on Embedded and Ubiquitous Computing, and the 2010 International Conference on VLSI Design. He has served as the Director of the Center for Embedded System-on-a-chip Design funded by New Jersey Commission on Science and Technology. He is the recipient of the AT&T Foundation Award and NEC Preceptorship Award for research excellence, NCR Award for teaching excellence, and Princeton University Graduate Mentoring Award. He received the Distinguished Alumnus Award from I.I.T., Kharagpur in 2014. He has co-authored or co-edited five books that include two textbooks: Testing of Digital Systems (Cambridge University Press, 2003) and Switching and Finite Automata Theory, 3rd edition (Cambridge University Press, 2009). He has also co-authored 15 book chapters and more than 410 technical papers. He has coauthored 14 papers that have won various awards, and another six that have received best paper award nominations. He has received 16 U.S. patents. He has served on the program committees of more than 150 conferences and workshops. His research interests include FinFETs, low power hardware/software design, computer-aided design of integrated circuits and systems, digital system testing, quantum computing and secure computing. He has given several keynote speeches in the area of nanoelectronic design and test.