Co-Design of Sparse Coding and Dictionary Learning for Real-Time Physiological Signals Monitoring

Kuan-Chun Chen, Ching-Yao Chou, An-Yeu (Andy) Wu
Graduate Institute of Electronics Engineering, National Taiwan University
No. 1, Sec. 4, Roosevelt Road, Taipei, 10617 Taiwan (R.O.C)
{kcchen, endpj}@access.ee.ntu.edu.tw, andywu@ntu.edu.tw

Abstract—Compressive sensing (CS) is a novel technique to reduce overall transmission power in wireless sensors. For physiological signals telemonitoring of wearable devices, chip area and power efficiency need to be considered simultaneously. There are many prior studies aim to develop algorithms that applied to CS reconstruction chips with reconfigurable architecture. However, representative dictionaries are also important when these CS reconstruction chips are verified in real-time physiological signals monitoring tasks. That is, a more representative dictionary can not only enhance the reconstruction performance of these chips but also alleviate memory overhead. In this paper, we apply the concept of co-design between sparse coding algorithms and learned dictionaries. We also explore the representativeness and compatibility of each learned dictionary. In addition, the computational complexity of each reconstruction algorithm is provided through simulations. Our results show that the dictionaries trained by fast iterative shrinkage-thresholding algorithm (FISTA) are more representative according to the quality of reconstruction for physiological signals monitoring. Besides, FISTA reduces more than 90% of the computational time compared with other hardware-friendly reconstruction algorithms.

Index Terms—Physiological signals telemonitoring, dictionary learning, signal reconstruction, compressive sensing

I. INTRODUCTION

With the increase of an aging population and chronic diseases, people become more aware of health issues. Due to the trend of healthcare systems moving toward patient-centered, many wearable and wireless biosensors have been invented these years [1]. It is believed that real-time medical telemonitoring will play an important role in the near future [2]. However, the telemonitoring system is facing the challenges of long-term monitoring and low data rate. Furthermore, most of the power in biosensors are dissipated during data transmission to wireless health hub [3]. As a result, it is necessary to reduce the amount of data to be transmitted and enhance the efficiency of transmission. To address these problems, compression techniques have been widely researched and discussed.

In [3], compressive sensing (CS) is able to extend 37.1% of a lifetime in sensor node which shows more lightweight sensors compared to Discrete Wavelet Transform (DWT) compression method. Therefore, CS is one of the most suitable choices for real-time telemonitoring tasks. The traditional CS framework can be divided into two parts, encoding process and decoding process. In CS encoding, the signal will be spread out to a smaller number of measurements through a sensing matrix. On the other hand, the decoding process, which is known as reconstruction process, aims to find a sparse solution to an underdetermined problem. However, for most real-time physiological signals monitoring, there are still some challenges need to be overcome.

1) Energy efficiency: For real-time analysis, energy efficiency is an important issue. As a result, a hardware-friendly sparse coding algorithm for VLSI implementation is essential in these scenarios.
2) Sparse characteristic: For most physiological signals, they are not sparse enough in time-domain, which leads to the failure of decoding. Therefore, we should find some proper transformations in order to create sparse characteristic.

Recent studies show that these physiological signals can be reconstructed properly if they have proper representation with respect to some bases or dictionaries [4]. Furthermore, dictionaries may explore the essence of data, and it is important in the domains of pattern recognition and medical diagnosis.

As shown in Fig. 1, different sparse coding algorithms create different dictionaries through dictionary learning phase. These dictionaries lead to different results in real-time inference which may cause the degradation of performance. That is, a more representative dictionary can not only enhance...
the performance in real-time inference, but also be more compatible with many existing CS reconstruction chips or hardware-friendly algorithms for biosensors. However, prior studies considered them separately. They mostly focus on different sparse coding algorithms analysis of CS reconstruction and dictionary learning methods [5], [6], [7], [8], [9]. They also focus on VLSI implementation for reconstruction algorithms [10], [11]. To our best knowledge, none of these studies have mentioned the co-designs between sparse coding algorithms and dictionary learning. Therefore, a comprehensive discussion about dictionary learning and inference is necessary for real-time CS based telemonitoring tasks. In this paper, we jointly considered the hardware-friendly sparse coding and dictionary learning. With applying different kinds of hardware-friendly sparse coding algorithms, we discuss the dictionary size and the representative capability of dictionaries. The main contributions of this paper are as follows,

1) **Co-design of dictionary learning and dictionary inference**: As mentioned above, the representativeness of each learned dictionary affects the capability of dictionary inference performance and compatibility. Hence, we explore the representativeness of dictionaries and focus on how dictionaries are trained. Besides, we aim to find out which dictionary is more compatible with existing CS reconstruction chips or hardware-friendly algorithms.

2) **Analysis of dictionary atom**: One tends to choose a smaller dictionary size and lower computational complexity with acceptable performance for hardware implementation. Therefore, we explore the memory overhead of dictionaries trained by several hardware-friendly sparse coding algorithms. Furthermore, we analyze the computational complexity of decoding process with these dictionaries.

The rest of paper is organized as follows: Section II will give an overview of the CS background and notations. Several reconstruction algorithms and dictionary learning methods are introduced in Section III. The experimental results are discussed in Section IV. Finally, we conclude this paper in Section V.

II. BACKGROUND AND NOTATIONS

**A. Compressive Sensing**

Assume that the original signal $x \in \mathbb{R}^N$ has a sparse representation with respect to basis $\Psi$ as

$$x = \Psi \alpha$$

where $\Psi$ is an $N \times L$ matrix ($L \leq N$), which is also known as a dictionary and $\alpha$ is an $L \times 1$ vector. Dictionary $\Psi$ can often be created by either predefined dictionary [12] or learned dictionary through several dictionary learning methods [13].

Compressive sensing (CS) is one of the compression techniques which can extend the lifetime of biosensors. In CS encoding, the original signal $x$ will be spread out to a smaller number of measurements $\hat{x}$ through a sensing matrix $\Phi$. Combining (1), it can be formulated as

$$\hat{x} = \Phi x = \Phi \Psi \alpha$$

where $\hat{x}$ is an $M \times 1$ and $\Phi$ is an $M \times N$ matrix ($M < N$). The compression ratio of the CS-based system is defined as $\frac{N}{M}$. In decoding process, receiver will reconstruct the transmitted measurements $\hat{x}$ by finding the sparsest solution $\alpha$ via reconstruction algorithms. The most intuitive way is to produce the sparsest solution to the $l_0$ minimization problem as

$$\min_{\alpha} ||\alpha||_0, \text{ s.t. } \hat{x} = \Phi \Psi \alpha$$

(3)

where $|| \cdot ||_0$ denotes the number of nonzero terms in its argument. At last, the reconstructed signal can be calculated by

$$\hat{x} = \Psi \alpha$$

(4)

where $\hat{x}$ is the reconstructed signal and has a dimension same as the original signal $x$.

**B. Related Works**

In [5], the work proposed and studied the use of alternating direction method (ADM) algorithms in CS but verified in synthetic database. [6] proposed a novel approach to estimate the relative electroencephalography (EEG) bands through compressive sensing. [7] demonstrated reconstruction performance using different sensing matrices and levels of sparsity with several sparse coding algorithms in physiological signals. [8], [9] mostly focus on sparse coding algorithms and dictionary learning methods. In [10] and [11], these studies aim to reduce hardware complexity with several orthogonal matching pursuit (OMP) algorithm modifications. However, these prior works focus on the performance of inference data or VLSI implementation. Co-design of dictionary learning and dictionary inference with different sparse coding algorithms are not mentioned. Therefore, we present a comprehensive discussion for several hardware-friendly CS reconstruction algorithms with learned dictionary involved.

III. CO-DESIGN OF RECONSTRUCTION ALGORITHMS AND DICTIONARY LEARNING METHODS

In this section, we first give an overview of reconstruction algorithms that attempt to solve (3). We also introduce several dictionary learning methods. At last, we mention the relations between sparse coding algorithms and learned dictionaries.

Sparse coding algorithms have been widely researched and discussed [8]. Only few of them are suitable for hardware implementation. It is computationally intractable to solve (3) in practice because its solution is NP-hard [14]. The most common way is to relax the constraint of (2) using $l_1$ minimization technique and to allow some error tolerance as

$$\min_{\alpha} \frac{1}{2} ||\hat{x} - \Phi \Psi \alpha ||^2_2 + \lambda ||\alpha||_1$$

(5)

Regularization parameter $\lambda > 0$ is used to provide a tradeoff between fidelity to the measurements and noise sensitivity. The first term is to seek the minimal error and the second $l_1$ term is used to induce sparsity.

The convex optimization problem (5) using $l_1$ minimization technique often offers an acceptable reconstruction quality. In order to solve (5), it is cast as several types of problems. One is to be cast as first order or second order programming and solve the problem via based pursuit denoising (BPDN). However, these types of methods often suffer from high computational complexity when the matrix data are dense or the signal dimension is significantly high.
Both energy efficiency and hardware friendly architectures are crucial when it comes to CS based telemonitoring tasks with biosensors. Therefore, this paper mostly focuses on hardware friendly reconstruction algorithms.

A. Hardware Friendly Reconstruction Methods

Since hardware is not as flexible as software, algorithms that have a regular procedure with acceptable reconstruction quality are often the best choice for hardware implementation. This type of algorithms will be the key for real-time applications of CS. Greedy algorithms, including matching pursuit based and gradient pursuit based, refine the current estimate iteratively and regularly. Matching pursuit based algorithms, such as orthogonal matching pursuit (OMP) [15], compressive sensing matching pursuit (CoSaMP) [16] and subspace pursuit (SP) [17] find the best matching projections during each iteration. Normal OMP procedure mainly includes three steps, pursuing process, LS estimation and residual update. In the most time-consuming process, pursuing process, OMP identifies the location of nonzero term iteratively. After OMP was proposed, there are several studies on improving OMP with better performance. However, they are not (Fig. 2).

The size of learned dictionary is no longer a square matrix which alleviates memory overhead compared to predefined dictionaries

There are several dictionary training approaches [13]. One way is to learn the characteristics of signals by machine learning methods. The other way utilizes $l_0$ or $l_1$ minimization techniques to induce sparsity for representations. For instance, online dictionary learning (ODL) [21], K-SVD [22] and method of optimal directions (MOD) [23]. Thus, the formulations of this type of methods are similar to (5) which can be solved by several sparse coding algorithms mentioned above. In the work of ODL, dictionary uses block-coordinate descent with warm restarts and creates two matrices that carry all the information from the past coefficient $\alpha$. The past coefficient $\alpha$ can be solved by sparse coding algorithms. In the work of MOD, the first step is to fix the dictionary $\Psi$ and find the sparse solution $\alpha$ via sparse coding algorithms. The solution can be formulated as

$$\min_{\alpha} \frac{1}{2} ||x - \Psi \alpha||^2_2$$

In the second step, MOD updates the dictionary as

$$\Psi = x\alpha^T(\alpha\alpha^T)^{-1}$$

while the sparse solution $\alpha$ is fixed. Within a number of iterative loops, learned dictionary $\Psi$ is created. The procedure of K-SVD is similar to MOD, the first step is to find the sparse representation with the dictionary fixed. However, in the second step, the dictionary update is performed one atom at a time which can be solved via an SVD decomposition.

C. Connections between Sparse Coding Algorithms and Dictionary Learning Methods

These dictionary learning methods require the sparse coefficient $\alpha$ to create the learned dictionary as shown in Fig. 2. During dictionary learning, we create the learned dictionary $\Psi$ via dictionary learning methods. After that, we find the sparse coefficient $\alpha$ via reconstruction methods based on the learned dictionary $\Psi$. Therefore, learned dictionaries will be influenced by different sparse coding algorithms, including training efficiency, inference performance, dictionary atom requirement and dictionary compatibility. That is, these dictionaries lead to different results in reconstruction tasks. A more representative dictionary can not only enhance the reconstruction quality but also be more compatible with many existing CS reconstruction chips or hardware-friendly algorithms for biosensors. We then perform several experiments to evaluate these dictionaries in the next section.
TABLE I
PRD AND CORRESPONDING QUALITY CLASS

<table>
<thead>
<tr>
<th>PRD</th>
<th>Reconstruction Signal Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ∼ 2%</td>
<td>&quot;Very good&quot; quality</td>
</tr>
<tr>
<td>2 ∼ 9%</td>
<td>&quot;Very good&quot; or &quot;good&quot; quality</td>
</tr>
<tr>
<td>≥ 9%</td>
<td>Not possible to determine the quality group</td>
</tr>
</tbody>
</table>

Fig. 3. Reconstruction quality (PRD) respects to dictionary atoms

IV. EXPERIMENTAL RESULTS

A. Experimental Settings

We use physiological signals to evaluate the performance of several reconstruction algorithms with dictionaries learned by MOD. Raw ECG signals were recorded from the intensive care unit (ICU) of stroke in National Taiwan University Hospital (NTUH). The sampling frequency of ECG signals is 512 Hz. Each sample includes 1 second ECG signal (input dimension \(N = 512\)). There are 2500 samples for dictionary learning and 1000 samples for inference. We first project the ECG signals by a sensing matrix \(\Phi\) to obtain the compressed signal \(\hat{x} = \Phi x\). Sensing matrix \(\Phi\) is designed as a random Gaussian matrix from a uniform distribution U(+1,-1). We set the compression ratio \(\frac{N}{M}\) as 4. Hence, the dimension of measurements \(\hat{x}\) is 128 \(M = 128\).

We first demonstrate the reconstruction performance of inference data. Afterwards, we show the representativeness of dictionaries trained by different sparse coding algorithms. To evaluate the performance of several reconstruction algorithms, reconstruction quality and computational complexity are calculated. We employ a widely used assessment index for reconstruction quality, namely, percentage root-mean-square difference (PRD) which is defined as [3]

\[
PRD = \frac{{\| x - \tilde{x} \|}_2^2}{{\| x \|}_2^2}
\]  

(8)

where \(x\) is the original signal and \(\tilde{x}\) is the reconstructed signal. Table I reports the resulting different quality classes and corresponding PRDs. The reconstruction quality is accepted if its PRD is smaller than 9% [24].

B. Experimental Results

1) Memory Overhead and Complexity Analysis: For hardware implementation, one tends to choose a smaller dictionary with acceptable reconstruction quality and reconstruction algorithms with lower computational complexity.

Fig. 3 shows the reconstruction quality respects to different dictionary atoms. Recall that dictionary \(\Psi\) is an \(N \times L\) matrix. Every reconstruction algorithm except SP can survive at \(L = 90\). Fig. 4 shows the computational complexity of each reconstruction algorithms, including OMP, FISTA, SGP and SP. FISTA achieves the lowest computational complexity, it reduces 97.9% of the computational time compared to OMP, 98.4% compared to SGP and 91.8% compared to SP (minimum dictionary atoms with acceptable PRD are chosen). We can observe that FISTA benefits from matrix inverse-free and gradient pursuit based procedure which achieves a better reconstruction quality with the lowest computational complexity.

2) Dictionary Representative Analysis: We evaluate the representativeness of dictionaries trained by different sparse coding algorithms. Since reconstruction quality is acceptable when the number of dictionary atom is larger than 90 (see Fig.3), we perform three simulations with \(L = 90, 150, 200\). Table II shows the reconstruction quality (PRD) with respect to different reconstruction algorithms and different dictionaries. The first column of the table is the size of dictionary and the second column indicates what sparse coding algorithms are applied during dictionary learning. It is shown that dictionaries trained by FISTA achieve a better reconstruction quality compared to other dictionaries. Besides, the lowest

TABLE II
DICTIONARY REPRESENTATIVE ANALYSIS

<table>
<thead>
<tr>
<th>Dictionary Atoms</th>
<th>Dictionary</th>
<th>OMP</th>
<th>FISTA</th>
<th>SGP</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>90</td>
<td>OMP</td>
<td>7.40</td>
<td>17.34</td>
<td>26.12</td>
<td>16.67</td>
</tr>
<tr>
<td></td>
<td>FISTA</td>
<td>7.21</td>
<td>6.87</td>
<td>8.48</td>
<td>8.32</td>
</tr>
<tr>
<td></td>
<td>SGP</td>
<td>7.22</td>
<td>7.60</td>
<td>8.48</td>
<td>8.32</td>
</tr>
<tr>
<td></td>
<td>SP</td>
<td>8.75</td>
<td>16.02</td>
<td>21.61</td>
<td>9.18</td>
</tr>
<tr>
<td>150</td>
<td>OMP</td>
<td>6.46</td>
<td>9.60</td>
<td>11.93</td>
<td>14.26</td>
</tr>
<tr>
<td></td>
<td>FISTA</td>
<td>5.50</td>
<td>4.32</td>
<td>7.87</td>
<td>4.65</td>
</tr>
<tr>
<td></td>
<td>SGP</td>
<td>6.16</td>
<td>5.20</td>
<td>6.81</td>
<td>6.10</td>
</tr>
<tr>
<td></td>
<td>SP</td>
<td>7.74</td>
<td>11.05</td>
<td>15.67</td>
<td>8.11</td>
</tr>
<tr>
<td>200</td>
<td>OMP</td>
<td>6.57</td>
<td>7.82</td>
<td>9.78</td>
<td>12.45</td>
</tr>
<tr>
<td></td>
<td>FISTA</td>
<td>5.68</td>
<td>4.05</td>
<td>6.49</td>
<td>4.55</td>
</tr>
<tr>
<td></td>
<td>SGP</td>
<td>5.98</td>
<td>4.69</td>
<td>6.57</td>
<td>5.65</td>
</tr>
<tr>
<td></td>
<td>SP</td>
<td>7.71</td>
<td>9.08</td>
<td>12.43</td>
<td>7.28</td>
</tr>
</tbody>
</table>

Bold numbers denote the lowest PRD respects to different dictionaries * denotes the lowest PRD in each group of dictionary atom
TABLE III
COMPARISON OF DIFFERENT SPARSE CODING ALGORITHMS

<table>
<thead>
<tr>
<th>Sparse Coding Algorithms</th>
<th>OMP</th>
<th>FISTA</th>
<th>SGP</th>
<th>SP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary memory overhead</td>
<td>90</td>
<td>90</td>
<td>90</td>
<td>&gt;90</td>
</tr>
<tr>
<td>Computational complexity</td>
<td>High</td>
<td>Lowest</td>
<td>High</td>
<td>Middle</td>
</tr>
<tr>
<td>Dictionary representativeness</td>
<td>Low</td>
<td>Highest</td>
<td>Middle</td>
<td>Low</td>
</tr>
</tbody>
</table>

PRD occurs when the reconstruction algorithm is FISTA. Therefore, dictionaries trained by FISTA are more representative and compatible among other dictionaries.

V. CONCLUSION

Most physiological signals require the assistance of dictionaries to have a sparse representation. Therefore, we should apply the concept of co-design between hardware-friendly sparse coding algorithms and learned dictionaries. We perform both dictionary atom and dictionary representative analysis. As shown in Table III, every reconstruction algorithms except SP can survive at L=90. However, FISTA shows a tremendous reduction in computational time due to relatively cheap matrix multiplication which enhances the feasibility of hardware design. In dictionary representative analysis, dictionaries trained by FISTA are more representative which shows the compatibility with existing CS reconstruction chips or hardware-friendly algorithms for wearable biosensors.

REFERENCES


