

A New Approach to Compressing ECG Signals with Trained Overcomplete Dictionary

SeungJae Lee, Jun Luan, and Pai H. Chou
Center for Embedded Computer Systems
University of California, Irvine, CA 92697-2625 USA
Email: leesj3@uci.edu, jluan1@uci.edu, phchou@uci.edu

Abstract—We propose a new ECG data compression algorithm based on a learned overcomplete dictionary to exploit the correlation between signals in adjacent heart beats. The learned overcomplete dictionary is constructed by K-SVD dictionary learning algorithm, after preprocessing and normalization of length and magnitude. Using the overcomplete dictionary, the proposed algorithm can find sparse estimation, which can represent the ECG signal effectively. Experimental results on MIT-BIH arrhythmia database confirms that our proposed algorithm has high compression ratio while minimizing data distortion.

I. INTRODUCTION

Electrocardiogram (ECG) data often require compression for storage, transfer over networks, and handling, especially given the trend towards miniaturization and wearable health-monitoring systems. ECG signal compression algorithm introduced in the past several decades can be lossless [1], [2], but lossy ones achieve better compression ratios by allowing small distortions in the parts of data that are not critical to the analysis of the signals. Lossy algorithms can be further categorized into direct signal compression and transformation compression. The former attempts to obtain redundancies on the signals to be compressed and extract meaningful parameters after analysis; the latter analyzes the signals in the transformed domain and retains the coefficients of the particular features in the data [3]–[6].

We propose a new approach to ECG signal compression based on heartbeat and dictionary learning. Our algorithm is divided into two stages. The training stage performs length normalization by interpolation to construct the overcomplete dictionary. During the compression/decompression stage, it calculates the optimal coefficients, which can capture most of the information content of the ECG signals. We compare our results with those of several well-known ECG compression algorithms using the MIT-BIH arrhythmia database [7].

This paper is organized as follows. In Section II, we introduce the background regarding sparse representation and dictionary learning. In Section III, we describe our proposed ECG compression algorithm using overcomplete dictionary constructed by K-SVD. Section IV presents experimental and comparison results using MIT-BIH arrhythmia database.

II. BACKGROUND

A. Sparse approximation

Sparse approximation is the representation that accounts for most of signal information with a linear combination of a small

number of elementary signals. Mathematically speaking, if the N -dimensional signal y is a sparse signal, which has k non-zero coefficients in the transformed domain, then the signal can be represented as

$$y = \Phi \alpha = \sum_{i=1}^k \alpha_i \phi_i \quad (1)$$

where Φ is called a dictionary, which is the set of elementary signals. Algorithms that find sparse approximation include matching pursuit (MP), orthogonal matching pursuit (OMP) [8], and least absolute shrinkage and selection operator (LASSO) [9].

B. Dictionary learning

The approximation performance in terms of the approximation quality and the sparsity of the sparse vector a depends on the dictionary as well as signal itself. If the dictionary contains well-estimated atoms or basis vectors, then the sparse vector will have a small number of non-zero elements. Universally used dictionaries, such as those for discrete cosine transform, Fourier transform, and Wavelet transform, are orthogonal dictionaries. They have mathematical simplicity and few redundancies to represent dictionaries themselves, but they are not suitable to represent signals with few redundancies. Due to the limitations of the orthogonal dictionaries, researchers proposed overcomplete dictionaries by data-driven learning. K-SVD [10] is a popular algorithms for constructing overcomplete dictionaries by learning. Given signals $Y = [y_1 y_2 \dots y_M] \in \mathbb{R}^{n \times N}$, the goal of K-SVD is to find the optimal dictionary $\Phi \in \mathbb{R}^{n \times K}$, where Φ is an overcomplete dictionary ($K > n$) and α is a sparse code of given input signals Y , keeping the sparsity constraints; $\|\alpha_i\|_0 \leq T_0, \forall i$.

$$\min_{\Phi, \alpha} \{\|y - \Phi \alpha\|_F^2\}, \text{ subject to } \|\alpha_i\|_0 \leq T_0, \forall i \quad (2)$$

III. PROPOSED ALGORITHM

Data compression can be viewed as the process of reducing the redundancies in a signal. One type of redundancies due to the quasi-periodic nature of the ECG signals is shown in the correlation between adjacent beats. To reduce the redundancy, we propose the beat-based data compression algorithm with an overcomplete dictionary. Our algorithm is composed training step and compression/decompression step. In training step, the

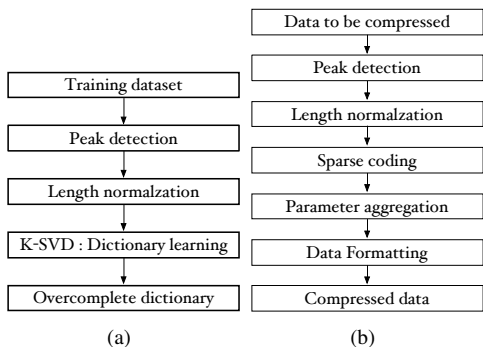


Fig. 1. (a) Dictionary construction process (b) compression process.

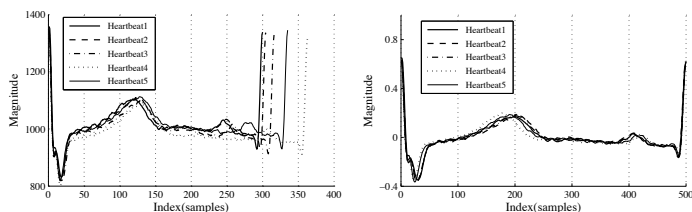


Fig. 2. (a) Original ECG signal (Record 231 from MIT-BIH Arrhythmia Database) (b) After normalization

algorithm constructs a set of training data for the learning dictionary, which will be constructed using K-SVD algorithm.

A. Dictionary Construction

Fig. 1(a) shows the flow of our dictionary-construction stage for the overcomplete dictionary from the given dataset. Although ECG signals are repetitive, they do not necessarily have a fixed period and morphology, even though they may have regular patterns. An ECG signal between two adjacent peaks of QRS complexes may show several morphological patterns with different lengths. To use this information for compression, our algorithm performs peak detection and length normalization.

The proposed algorithm uses the Pan and Tomkins QRS peak detection method [11] to extract the patterns between the peaks of two adjacent QRS complexes. After the detection, length and magnitude normalizations are needed to regularize the peak-to-peak ECG signal for constructing a set of training data. To remove the DC component of the signals, the algorithm makes the extracted pattern zero-mean. Fig. 2(a) shows that each ECG signal has a different length. Fig. 2(b) depicts the result of normalization process.

After normalization, the k^{th} peak-to-peak data d_k , whose length is not equal to n , could be represented as an n -dimensional column vector y_k . A set of these normalized vectors could be presented to $Y = [y_1 y_2 \dots y_k]$, called the training dataset. Using Y , the optimized dictionary to represent the data under the sparsity constraint is obtained by K-SVD dictionary learning.

B. Compression

Our proposed compression procedure is illustrated in Fig. 1(b). It compresses beat-based data, so the target data

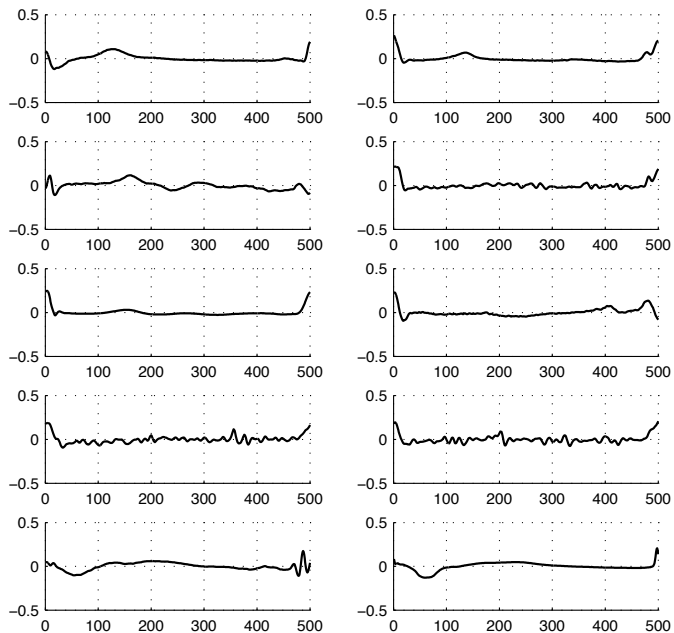


Fig. 3. Atoms of Learned Dictionary by K-SVD

should be extracted before compression. During dictionary construction, QRS complexes should be detected to distinguish adjacent beats. Then, X , which is a selected target beat to be compressed, should be length-normalized. To recover the signal in the decompression step, the length of the beat should be stored.

In the sparse coding stage, the algorithm finds the sparse vector using the given overcomplete dictionary. From Eq. (1), the sparse code in α is not a unique vector, since the given dictionary is overcomplete. To obtain the appropriate coefficients, which can represent the signal with a small amount of error, sparsity constraint should be included in the equation. Sparse solutions with overcomplete dictionary can be found using MP, OMP, and LASSO, as discussed in Section II. We use OMP, a greedy algorithm, for its simplicity and efficiency.

In the parameter aggregation stage, the proposed compression algorithm needs to store information about the heartbeats. The beat-related information to be stored is in terms of the average value of the signal in a beat and the length of the beat. If the dimension of the constructed dictionary is $n \times K$, then the sparse vector α is $K \times 1$. Due to the overcomplete requirements, K is much larger than n . Under the sparsity constraint, most of the elements in sparse vector α are zero. Only a small number of elements have non-zero coefficients. To reconstruct the sparse vector α during decompression, the index of the non-zero elements should be stored.

C. Decompression

The decompression step is exactly the reverse procedure of compression except for the sparse coding stage. For the decompression, the sparse decoding stage is replaced by the sparse coding. Since the elements of sparse vector means the

score of respective atoms of the given dictionary, the decoding merely requires the linear combination of the relevant atoms. After sparse decoding, the algorithm restores the original length of signal using stored length information.

IV. EXPERIMENTAL RESULTS

For comparing the results of the proposed algorithm with those of conventional ECG compression methods, we have chosen the MIT-BIH arrhythmia database [7] as the test signals, which are widely used for evaluating the performance of ECG compression algorithms. The database provides 48 sets of ECG signals, which are all two-lead ECG data. Each lead is recorded at 360 samples/sec with 11-bit resolution. To quantify the performance of the proposed algorithm, we employ three widely used metrics: compression ratio (CR), percentage root-mean-squared distortion (PRD), and quality score (QS).

A. Measurement of Performance

Compression ratio (CR) is defined as

$$CR = \frac{\text{the number of bits of original signal}}{\text{the number of bits of compressed signal}} \quad (3)$$

Another widely used metric is percentage root-mean-squared distortion (PRD), which shows the reconstruction error as a percentage. It is defined as

$$PRD = \sqrt{\frac{\sum_{i=1}^n (x_i - \hat{x}_i)^2}{\sum_{i=1}^n x_i^2}} \times 100\% \quad (4)$$

where n is the number of samples, and x_i and \hat{x}_i are the original data and the reconstructed data from our proposed algorithm, respectively.

The quality score (QS) [12] is the ratio between the CR and PRD:

$$QS = \frac{CR}{PRD} \quad (5)$$

High value of QS means good performance on lossy compression. It is useful to choose the best method while taking the compression reconstruction errors into consideration as well.

B. Parameters for Dictionary Construction

Since our proposed algorithm is learning-based, we should construct data sets for training purpose as explained Section III. We have chosen 24 out of 48 signals from the MIT-BIH arrhythmia database. Of the 48 signals, 24 are utilized by our K-SVD based algorithm to construct the dictionary, while the rest are used to evaluate the performance of our proposed algorithm. Table I depicts the signals included in training dataset, totaling 51,492 beats per lead.

Parameter selection is a key for good quality of dictionary construction. Several user-specific parameters on the procedures include sparsity, the size of dictionary, and the size of length normalization. The proposed algorithm uses $T_0 = 10$ as sparsity constraint for dictionary construction. To satisfy the overcomplete requirement, we set the dictionary size to 1024. Furthermore, the algorithm normalizes the peak-to-peak

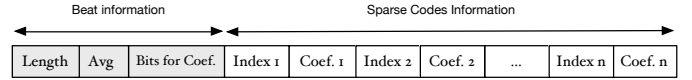


Fig. 4. Compressed data format

TABLE I
DATA SET FOR TRAINING AND TESTING FROM MIT-BIH ARRHYTHMIA DATABASE

Records for Training	Records for Testing
100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 111, 112, 113, 114, 115, 116, 118, 119, 121, 122, 123, 124, 200, 201	117, 202, 203, 205, 207, 208, 209, 210, 212, 213, 214, 215, 217, 219, 220, 221, 222, 223, 228, 230, 231, 232, 233, 234

signals to 500 samples by linear interpolation. The size of the normalized peak-to-peak signals should be larger than the original peak-to-peak signals for preserving the signals after the processing.

The magnitudes of the coefficients are determined by the dictionary. The coefficients are determined by projecting the normalized data on the atoms of the dictionary. Since the magnitude of each atom of the dictionary is unit-length, each element in the atoms dictionary is quite small. More bits should be allocated to represent the coefficients; but to avoid over-allocation, we scale up the dictionary. Experimentally, we use 10 as a scaling factor.

C. Data format

Fig. 4 shows our proposed data format after parameter aggregation for representing one heartbeat. In the beat information header, we allocate 11 bits for representing the average peak-to-peak value and 9 bits for the length. Due to the total dictionary size, we allocate 10 bits for each index, which is the location of the atom with respect to the coefficient value. The size of the coefficient field varies between 8-10 bits in these experiments.

D. Evaluation

To recall, our metrics are CR, PRD, and QS. For the purpose of evaluating performance, the first 10 heart beats of the signal are used to the compression. Table I shows the training dataset for constructing dictionary and the testing dataset for evaluation.

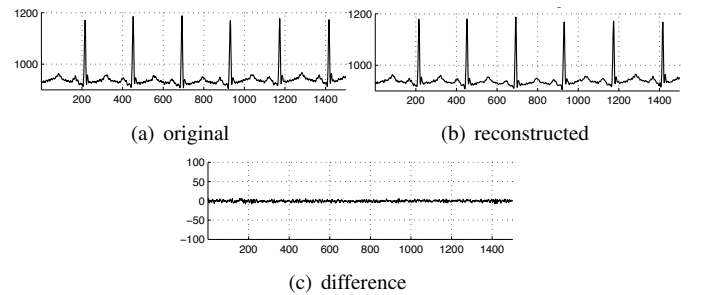


Fig. 5. Signal reconstruction with the proposed algorithm(Record 205 from MIT-BIH arrhythmia database)

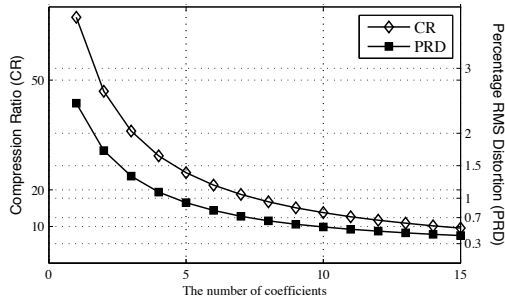


Fig. 6. Average of CR and PRD over testing dataset(MIT-BIH arrhythmia database)

TABLE II
PERFORMANCE COMPARISON WITH CONVENTIONAL ALGORITHMS

Algo-rithm	SPIHT [4]	AZTEC [13]	CORTES [14]	Hilton [5]	Djohan [15]	Fira [12]	Ours
PRD	1.18%	28%	7%	2.6%	3.9%	0.61%	0.55%
CR	8:1	10:1	4.8:1	8:1	8:1	12.74:1	13.79:1
QS	9.3	3.57	0.68	3.076	2.05	20.88	24.75

Fig. 5 shows that the reconstructed signal contains small deviations from the original signal. The original signal is Record 205 from MIT-BIH arrhythmia database. With Record 205, $CR = 11.23$, $PRD = 0.30\%$, $QS = 37.43$ are achieved. The highest compression ratio ($CR = 18.27$, $PRD = 0.69\%$) is achieved for Record 117. The lowest compression ratio ($CR = 7.99$, $PRD = 0.75\%$) is achieved for Record 213. The proposed algorithm has high CR in the range of 7.99-18.27 and low PRD in the range of 0.28-1.52%.

The CR and PRD have close relationship in lossy compression. In general, high CR implies high PRD, and low CR leads to low PRD. Since the proposed algorithm is a beat-based compression algorithm, the CR is determined by the number of coefficients that represent a heart beat. Fig. 6 shows the relationship between CR and PRD over entire testing dataset.

Table II shows the comparison results with the conventional methods, such as SPIHT [4], AZTEC [13], CORTES [14], Hilton [5], Djohan [15] and Fira [12]. Our proposed algorithm, which uses 10 coefficients to represent each heart beat, has better PRD performance than other conventional methods do. Our algorithm has an average PRD of 0.55%, CR of 13.79:1, and QS of 24.75 over the testing dataset.

V. CONCLUSIONS

We propose a new two-stage algorithm for ECG signal compression based on dictionary learning from training datasets. The dictionary construction stage uses the K-SVD algorithm with given set of training data, which is zero-mean and length-normalized. The compression/decompression stage first extracts the peaks of QRS complexes by using Pan and Tompkins method and then finds the best-matched sparse codes with the trained dictionary. The algorithm has been tested with

MIT-BIH arrhythmia database and compared with existing methods in terms of the compression ratio (CR), percentage of root mean squared distortion (PRD), and quality score (QS). Our algorithm shows good compression performance with low distortion. Since the algorithm is a learning-based method, the performance can improve with the training dataset.

VI. ACKNOWLEDGEMENT

Research reported in this publication was supported by National Heart, Lung and Blood Institute (NHLBI) of the National Institutes of Health under award number R41HL112435. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

REFERENCES

- [1] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, no. 6, pp. 8–19, 1984.
- [2] British Standards Institute Staff, *Health Informatics. Standard Communication Protocol. Computer-Assisted Electrocardiography*. B S I Standards, 2005. [Online]. Available: <http://books.google.com/books?id=u8hbAAAACAAJ>
- [3] S. Olmos, M. Millan, J. Garcia, and P. Laguna, "ECG data compression with the Karhunen-Loeve transform," in *Computers in Cardiology*. IEEE, 1996, pp. 253–256.
- [4] Z. Lu, D. Y. Kim, and W. A. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 47, no. 7, pp. 849–856, 2000.
- [5] M. L. Hilton, "Wavelet and wavelet packet compression of electrocardiograms," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 5, pp. 394–402, 1997.
- [6] G. Nave and A. Cohen, "ECG compression using long-term prediction," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 9, pp. 877–885, 1993.
- [7] 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, 2000.
- [8] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [9] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [10] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [11] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, no. 3, pp. 230–236, 1985.
- [12] C. M. Fira and L. Goras, "An ECG signals compression method and its validation using NNs," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 4, pp. 1319–1326, 2008.
- [13] J. Cox, F. Nolle, H. Fozzard, and G. Oliver, "AZTEC, a preprocessing program for real-time ECG rhythm analysis," *IEEE Transactions on Biomedical Engineering*, no. 2, pp. 128–129, 1968.
- [14] J. P. Abenstein and W. J. Tompkins, "A new data-reduction algorithm for real-time ECG analysis," *IEEE Transactions on Biomedical Engineering*, no. 1, pp. 43–48, 1982.
- [15] A. Djohan, T. Q. Nguyen, and W. J. Tompkins, "ECG compression using discrete symmetric wavelet transform," in *Proc. IEEE 17th Annual Conference on Engineering in Medicine and Biology Society*, vol. 1. IEEE, 1995, pp. 167–168.