High-rate compression of ECG signals by an accuracy-driven sparsity model relying on natural basis

Giuliano Grossi, Raffaella Lanzarotti, Jianyi Lin[*]

Department of Computer Science, University of Milan
Via Comelico 39, 20135 Milan, Italy

Abstract

Long duration recordings of ECG signals require high compression ratios, in particular when storing on portable devices. Most of the ECG compression methods in literature are based on wavelet transform while only few of them rely on sparsity promotion models. In this paper we propose a novel ECG signal compression framework based on sparse representation using a set of ECG segments as natural basis. This approach exploits the signal regularity, i.e. the repetition of common patterns, in order to achieve high compression ratio (CR). We apply *k*-LiMapS as fine-tuned sparsity solver algorithm guaranteeing the required signal reconstruction quality (PRD). Extensive experiments of our method and of four competitors (namely ARLE, Rajoub, SPIHT, TRE) have been conducted on all the 48 records of MIT-BIH Arrhythmia Database. Our method achieves average performances that are 3 times higher than the competitor results. In particular the compression ratio gap between our method and the others increases with the PRD growing.

Keywords: ECG compression, sparse representation, PRD guaranteed, high compression ratio, cardiac arrhythmia

1. Introduction

ECG signals are essential in the diagnosis of heart diseases. Their acquisition consists in applying about 5-7 electrodes on the body and, for long recordings, the signals can be acquired even over 24 h, thus producing a large volume of data. In addition, the progress in technology allows an improvement of the acquisition precision (e.g. sampling rate, resolution), leading to a further grow of the amount of digital ECG data. Considering the limited amount of storage space and bandwidth, an efficient ECG data compression plays a crucial role in the implementation of electrocardiogram acquisition systems. Clearly, a constraint of the compression solution is to maintain the ECG characteristics such as heart rate, P-wave, QRS-complex, T-wave, etc., which constitute important evidences for medical diagnosis [27].

The pioneer ECG compression methods aimed at analyzing and removing the redundancy from the time-domain signals to achieve data compression (e.g. AZTEC and CORTES) [12]. Such methods produce small reconstruction error but also low compression. More recently, transform-based methods have demonstrated their superiority in compressing EGC signals, converting the signal to another representation more suitable to detect and remove the redundancies. Among them, the wavelet transforms [15, 25, 13] have been extensively used because of their properties of good location in time and frequency domains. Nowadays a new paradigm, namely Compressed Sensing (CS), has gained increasing attention, having proven its effectiveness [16, 14]. Researches in this field have been focused on two key aspects, namely the sparse coding techniques and the dictionary construction. The first concerns with the algorithm to find the sparse representation (SR) of a sample, as for instance Basic Pursuit (BP) [6], Orthogonal Matching Pursuit (OMP) [19], Least Absolute Shrinkage and Selection Operator (LASSO) [26], *k*-LiMapS[1], each

[*]Corresonding author

Email addresses: grossi@di.unimi.it (Giuliano Grossi), lanzarotti@di.unimi.it (Raffaella Lanzarotti), jianyi.lin@unimi.it (Jianyi Lin) one adopting a peculiar metric that conditions both the efficacy and the efficiency of the method [8]. The second aspect concerns with the construction of a proper dictionary to be referred by the SR algorithm when searching sparse solutions. As deepened in [24], earlier works made use of Fourier or wavelet dictionaries, which are simple to use and perform well. However, these dictionaries are not well suited for representing complex natural data for which major flexibility is required. To overcome this issue, in the ECG domain, dictionaries collecting chunks of records were developed [9].

In this paper we propose an ECG compression method framed in the sparse representation field, whose early stage has been presented in [3]. In such a work the problem of succinctly representing heart beats is recast into a regularization problem with approximate constraints. These constraints are expressed by a dictionary built on normalized initial transient of the original ECG record. In particular, for each patient his/her own dictionary is built, collecting and aligning the R-R intervals of the first part of the ECG signals (about 5-10 min), while compressing the remaining ones. The problem is hence tackled by the *k*-LiMapS sparsity solver, which is essentially an iterative scheme aimed at finding the sparsest solution of the dictionary-based linear system. The present contribution introduces major improvements in the core of the *k*-LiMapS algorithm toward two directions: on the one hand the reconstruction quality (PRD) requirements are intrinsically included into the sparsity recovery scheme, thus resulting into a PRD guaranteed method; on the other hand we introduce at the end of the sparsity recovery a step of least-squares projection yielding the optimal point within the subspace spanned by the atoms selected by *k*-LiMapS. As minor improvement, we have introduced Tikhonov regularization into the sparsity solver in order to make it more robust against zero-padding normalization. Moreover, we tackle the rare cases of non-sparsifiable ECG segments, i.e. segments requiring too many dictionary atoms for their reconstruction. To this end we resort to a backup procedure that uses standard wavelet transforms.

The performance of the proposed algorithm is evaluated on the whole MIT-BIH Arrhythmia Database (48 records) in terms of Compression Ratio (CR) and Percentage Root-mean-square Difference (PRD). Furthermore, direct comparisons with four state-of-the-art ECG compression methods, namely ARLE [4], Rajoub [22], TRE [5], SPIHT [15], are conducted. In summary, the results show that, for low PRD values, our method achieves superior CR with respect to the competitors on three quarters of the dataset, while for higher PRD values, our method considerably outperforms the others on the whole dataset.

The rest of the paper is organized as follows: in Section 2 we describe the proposed framework; in Section 3 we present the details of our sparsity model; experimental analysis and comparative results are shown in Section 4; finally, in Section 5 we draw some conclusions and sketch possible future works.

2. The framework

The overall ECG compression method proposed here is sketched in the block diagram of Fig. 1.
It consists of four stages, described below:

i) signal preprocessing through standard filtering for wandering removal, R-peaks detection and normalization based on zero-padding of centered RR-segments

ii) dictionary construction over natural basis extracted from the initial transient of the normalized record

iii) online sparse decomposition via the sparsity solver *k*-LiMapS combined to the least-squares projection (LSP) and resorting to the DWT in case of either long or non-sparsifiable segments

iv) quantization and compression of the coefficients carried out both by the sparsity process and (possibly) by DWT using the arithmetic coding.

2.1. Signal Preprocessing

First of all, the original ECG record is filtered by both a high-pass filter and a notch filter in order to remove signal wandering due to human breathing and standard noise. Then the ECG record has to be segmented and normalized in order to obtain a collection of basic waveforms used both to build a natural-basis dictionary (see Sec. 2.2) and to feed the sparsification process (see Sec. 2.3). The segmentation step consists in isolating the R-peaks of the ECG record by

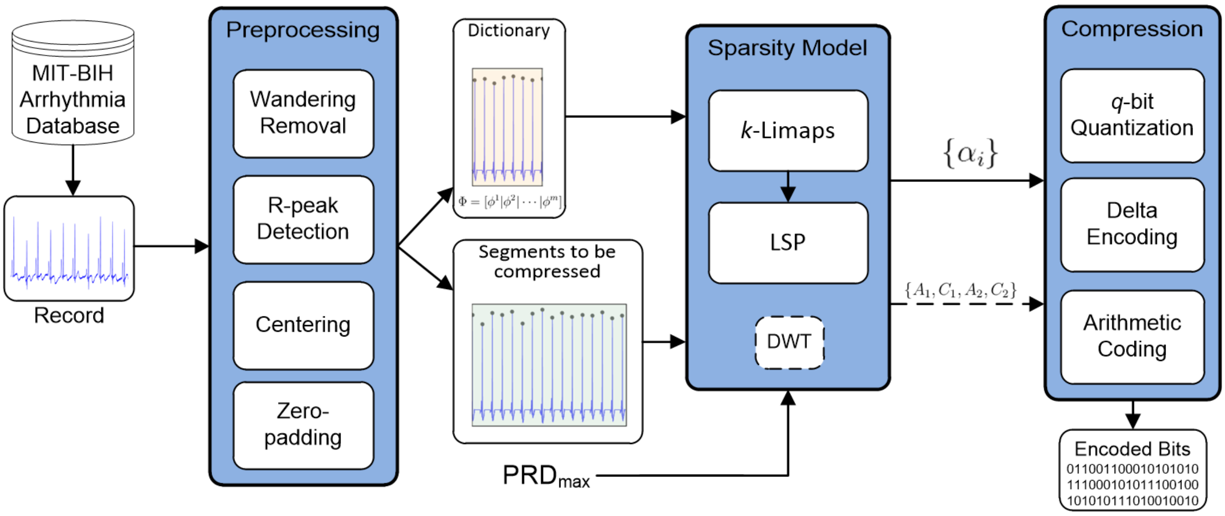

Figure 1: Block diagram of the ECG signal compression process, showing the principal elements of our framework.

a standard peak detection technique [17], so as to subdivide the entire signal into RR-segments (or simply segments), denoted by s_i; clearly every segment s_i has a different length $n_i = |s_i|$.

The preprocessing step aims at obtaining normalized segments $S = \{y_1, ..., y_M\}$ with zero mean (centering) and a common length n sufficiently large. Specifically, it is performed by subtracting from each segment s_i its average value $\mu(s_i)$ and then extending the obtained zero-mean segment by inserting $z_i = n - n_i$ many zeros in the middle (zero-padding). To fix the length n, since a heart beat occurs in about a second, a reference parameter could be the sampling frequency F_s. However, in order to deal with high variability in the heart beat intervals, we set $n = \gamma F_s$, where $\gamma > 1$ is a constant that does not affect the sparsity model, provided that it is chosen sufficiently large. Occasionally, there may happen that few RR-segments still exceed n, and in this case the segment is processed separately by means of a simple backup procedure (see Sec. 3.3).

This preprocessing allows our method to achieve higher compression ratios than aligning the R-peaks in the center, and interpolating the block to attain a desired length as in [20].

2.2. *Design of the dictionary*

Aim of the sparse modelling applied to the ECG compression task is to capture the essential characteristics of ECG signals with only a few coefficients. Another desired property is to provide wide flexibility so as to adapt the coding process to both near-periodic and highly irregular signals.

The requirements mentioned above are the reasons behind the sparse model adopted here which relies on a dictionary $\Phi \in \mathbb{R}^{n \times m}$ naturally extracted from the signal at hand. This is done by concatenating in a column-wise fashion a suitable amount of normalized RR-segments picked from an initial transient of about 5-10 minute duration. Since sparsity modelling requires overcomplete dictionaries, the number of columns of the matrix Φ must respect the constraint $m > n = \gamma F_s$. In other words, the dictionary $\Phi = [\phi^1|\phi^2|\cdots|\phi^m] \in \mathbb{R}^{n \times m}$ is a collection of atoms $\phi^j \in \mathbb{R}^n$, $j = 1, ..., m$, arranged as columns and corresponding to the m normalized RR-segments taken from the initial transient interval of the record.

Once the dictionary is created, the remaining normalized segments $S = \{y_1, ..., y_M\} \subset \mathbb{R}^n$ of the record are compressed using the sparse representation model, as formulated in the following subsection.

We observe that, this dictionary design approach is convenient when the ECG data are recorded for a long duration with respect to the transient interval allocated to the dictionary, that is $M >> m$.

2.3. *Sparse Approximation*

A vector $\alpha \in \mathbb{R}^m$ is said to be *k-sparse* iff its pseudo-norm $||\alpha||_0 := \#\,\mathrm{supp}(\alpha) = \#\{i : \alpha_i \neq 0\} \leq k$, i.e. the number of non-null elements is at most k, which is called the sparsity (or sparsity level) of α. Let $\Sigma_k = \{x \in \mathbb{R}^m : \#\,\mathrm{supp}(x) \leq k\}$

denotes the collection of k-sparse vectors in $\mathbb{R}^m$. A *k-sparse representation* of any vector $y \in \mathbb{R}^{n\times 1}$ by the dictionary Φ is the equality

$$y = \alpha_1\phi^1 + \alpha_2\phi^2 + \cdots + \alpha_m\phi^m \tag{1}$$

with suitable $\alpha \in \Sigma_k$, or equivalently in matrix form $\Phi\alpha = y$ s.t. $||\alpha||_0 \leq k$. This means that y can be represented in a subspace generated by k atoms of Φ. Clearly, when $k \geq n$ the k-sparse representation always exists for a full-rank matrix Φ, whilst if a k-sparse α exists with a small $k \ll n = |y|$, it can be considered as a compact representation of the signal $y \in \mathbb{R}^{n\times 1}$, which essentially turns out to be useful for compression purposes, since y can be exactly reconstructed from α.

In fact, in many applications like signal compression, a reconstruction error is accepted, and hence instead of requiring the equality (1), one fixes an $\varepsilon > 0$ and searches for the sparsest vector $\alpha \in \mathbb{R}^m$ that satisfies the error constraint $||\Phi\alpha - y|| \leq \varepsilon$, where $|| \cdot ||$ denotes the Euclidean norm. This leads to formulate the classical *sparsest approximation problem* [8]:

$$\hat{\alpha} = \underset{\alpha\in\mathbb{R}^m}{\operatorname{argmin}} \ ||\alpha||_0 \qquad \text{subject to} \qquad ||\Phi\alpha - y|| \leq \varepsilon \tag{2}$$

In practice, this problem is well-known to be NP-hard [18]. As mentioned in the introduction, there are various heuristics for finding an approximate solution to this problem. Briefly, the BP method finds signal representations in overcomplete dictionaries by convex optimization, obtaining the decomposition that minimizes the ℓ_1-norm of the coefficients occurring in the representation. The OMP method [19] consists in a greedy search for the subspace of $\mathbb{R}^n$ generated by progressively adding more atoms such that the residual given by the projection in the orthogonal complement space is minimum at each iteration. The LASSO algorithm [26] minimizes the residual sum of squares trying to select the atoms subject to the sum of absolute values of the coefficients being less than a constant. The k-LiMapS algorithm, proposed in [1], is a sparsity recovery method that consists in a fixed-point iteration scheme that promotes the sparsity of partial solutions by suitable non-linear projections.

In this work we develop a framework based on the k-LiMapS algorithm that has already proven its effectiveness in ECG compression [3] and also in the field of face recognition [2]. We stress that here we use an enhanced version of k-LiMapS as described in Sec. 3.

2.4. Encoding

After constructing the dictionary Φ, every normalized segment $y_i \in S$, corresponding to the RR-segment s_i to be compressed, is represented by a sparse vector $\hat{\alpha}_i \in \mathbb{R}^m$, solving problem (2) as describe in Sec. 3. The collection of these vectors has to be quantized and then compressed. The quantization is carried out as follows. Given the coefficient vectors $\hat{\alpha}_1, ..., \hat{\alpha}_L \in \mathbb{R}^m$ corresponding to the $L \leq M$ segments $s_1, ..., s_L$ well-sparsifiable by k-LiMapS, let $k_i = ||\hat{\alpha}_i||_0$, $i = 1, ..., L$, be their number of non-null coefficients. Hence, we have to quantize a sequence of $K = \sum k_i$ non-null coefficients that we denote with $c_1, ..., c_K \in \mathbb{R}$. To this end, we use a delta encoding with a uniform q-bit quantization with suitable $q \in \mathbb{Z}$, so that the set of c_j's is encoded by the level-indices of the quantized deltas

$$\Delta_j^Q = \left[\frac{\Delta_j - \Delta_{\min}}{\Delta_{\max} - \Delta_{\min}}(2^q - 1)\right] \in \{0, 1, ..., 2^q - 1\}$$

where $\Delta_j = c_j - c_{j-1}$, and $\Delta_{\min}$, $\Delta_{\max}$ are respectively the minimum and the maximum among the Δ_j's. Clearly, the coefficients $c_1, \Delta_{\min}, \Delta_{\max}$ has to be stored apart. In addition, we need to quantize the mean value $\mu_i = \mu(s_i)$ of each segment s_i.

Together with the non-null coefficients of the vectors $\hat{\alpha}_i, i = 1, \ldots, L$, we have to store *i*) their support, i.e. the relative positions within each vector, *ii*) the number $z_i = n - n_i$ of zeros inserted during padding and *iii*) their sparsity level k_i. These elements are stored using the delta encoding. Notice that with all the quantized information listed up to this point, it is straight-forward to compute the approximated reconstruction $\tilde{\alpha}_i$ of the $\hat{\alpha}_i$, and hence the normalized segments $\tilde{y}_i = \Phi\tilde{\alpha}_i$. In order to compute the reconstructed segment $\tilde{s}_i$, that is an error-controlled approximation of s_i, it is sufficient to remove z_i padded coefficients and re-centering with μ_i from $\tilde{y}_i$.

As compression technique we use the arithmetic coding [7], mainly for its capability to better cope with the symbol probabilities arising from the encoding process described above.

In order to measure the effectiveness of our ECG compression technique on a given record $x = (s_1, ..., s_M)$ we use the standard Compression Ratio (CR) defined as follows:

$$\mathrm{CR} = \frac{\#\text{ bits of } x}{\#\text{ bits of } C(x)}$$

where $C(x)$ is the sequence of compressed codewords for x produced after the arithmetic coding. The number of bits of $C(x)$ obviously amounts to the arithmetic codeword lengths, while if a record x has N_s samples with a q_R-bit quantization depth, the number of bits for x is simply calculated as $N_s \cdot q_R$.

3. The sparse decomposition stage

In this section, after having briefly recalled the core of the adopted sparse decomposition method (namely k-LiMapS), we highlight the four key contributions introduced in this work. The first point is the integration with the least squares optimization to find the solutions having minimum errors within the subspace spanned by $\hat{\alpha}$'s support; the second aspect concerns with the iterative search procedure to get the sparsest solution with guaranteed PRD; thirdly, the method is integrated with a Backup procedure to manage effectively non-sparsifiable segments; the last point consists in the implementation of the Tikhonov regularization to face the problem of ill-conditioned dictionaries.

3.1. The k-LiMapS *algorithm and the least squares minimization step*

- The k-LiMapS algorithm

  The main goal of the proposed framework is to gain compression by promoting the sparsity of the solutions to problem (2). To this aim we use the k-LiMapS method, that is a thresholding-based iterative process for model selection as described in details in [1]. It essentially relies on a parametric family of uniformly Lipschitzian nonlinear mappings $F_\lambda : \mathbb{R}^m \to \Sigma_k$ such that $||F_\lambda(x)|| < ||x||$, $\forall \lambda > 0$ and where the parameter λ controls the shrinking effects, i.e. leads the search towards the sparsest solutions. After the shrinking, the pseudo-inverse[1] $\Phi^\dagger$ of the dictionary Φ is used to restore the feasibility at each iteration step.

  Thus, for a given normalized segment y_i and a fixed sparsity level k, we have the fundamental sparsification statement:

  $$\hat{\alpha}_i \leftarrow k\text{-LiMapS}(y_i, \Phi, \Phi^\dagger, k). \tag{3}$$

  The aim of step (3) is to identify the k most suitable atoms in Φ giving a good approximate sparse solution $\hat{\alpha}_i$ s.t. $\Phi\hat{\alpha}_i \approx y_i$.

- The least squares optimization

  In order to reduce the error $||\Phi\hat{\alpha}_i - y_i||$, we introduce a least-squares projection in the subspace spanned by the k atoms selected by $\hat{\alpha}_i$, thus identifying the closest representation preserving the sparsity level reached in (3). In other words, by denoting with $\mathcal{S}_i = \mathrm{supp}(\hat{\alpha}_i)$ the support of $\hat{\alpha}_i$, let $\Phi_{\mathcal{S}_i}$ be the submatrix collecting the atoms in Φ whose column-indices are in $\mathcal{S}_i$. The solution of the problem

  $$\beta_i^* = \underset{x \in \mathbb{R}^k}{\mathrm{argmin}} \; ||\, y_i - \Phi_{\mathcal{S}_i} x \,||$$

  corresponds to the point closest to $\hat{\alpha}_i$ in the subspace spanned by its atoms, with respect to the Euclidean distance. The effective computation can be performed through the projection given by the pseudo-inverse of the matrix $\Phi_{\mathcal{S}_i}$, i.e.

  $$\beta_i^* = \left(\left(\Phi_{\mathcal{S}_i}^T \Phi_{\mathcal{S}_i}\right)^{-1} \Phi_{\mathcal{S}_i}^T\right) y_i = \Phi_{\mathcal{S}_i}^\dagger y_i \in \mathbb{R}^k$$

[1] Give a matrix A, its Moore-Penrose pseudo-inverse is uniquely identified by $A^\dagger = (A^T A)^{-1} A^T$ or $A^\dagger = A^T (AA^T)^{-1}$ according to whether $A^T A$ or AA^T is invertible.

Then we build a new vector $\alpha_i^* \in \mathbb{R}^m$ with the same support of $\hat{\alpha}_i$, assigning to its j-th non-null element the j-th element of β_i^*. This step is summarized by the statement:

$$\alpha_i^* \leftarrow \mathrm{LSP}(\Phi, \hat{\alpha}_i). \tag{4}$$

3.2. Sparsity led by PRD guaranteed

Driven by the accuracy desired for the reconstructed signal, we derive a method that tunes the sparsity level k in order to keep the error under control. To evaluate the quality of the reconstruction, we use the classical measures, namely the percentage root-mean-square difference (PRD) and its normalized version (PRDN):

$$\mathrm{PRD}(x, \hat{x}) = \frac{||x - \hat{x}||}{||x||} 100, \qquad \mathrm{PRDN}(x, \hat{x}) = \frac{||x - \hat{x}||}{||x - \mu(x)||} 100.$$

where $\mu(x)$ is the temporal mean of the signal x. Note that, in order to guarantee a maximum reconstruction error $\mathrm{PRD}_{\mathrm{max}} > 0$ for a given segment s_i, the constraint $\mathrm{PRD}(s, \hat{s}) \leq \mathrm{PRD}_{\mathrm{max}}$ can be rewritten as

$$||s - \hat{s}|| \leq \frac{||s||}{100} \mathrm{PRD}_{\mathrm{max}}.$$

Since $||s - \hat{s}|| \leq ||y - \hat{y}|| = ||y - \Phi\hat{\alpha}||$, a solution $\hat{\alpha}$ of the problem (2) with $\varepsilon := \frac{||s||}{100}\mathrm{PRD}_{\mathrm{max}}$ is sufficient to meet the requirement for the reconstruction error of $\hat{s}$, i.e. the problem turns out to be

$$\hat{\alpha} = \underset{\alpha \in \mathbb{R}^m}{\operatorname{argmin}} \; ||\alpha||_0 \quad \text{subject to} \quad ||\Phi\alpha - y|| \leq \frac{||s||}{100} \mathrm{PRD}_{\mathrm{max}}$$

This problem suggests that the hardest part of the combinatorial search for an optimal ℓ_0 solution is committed to the sparsity solver algorithm, which in turn means finding the subspace on which the sparsest solution satisfying the error requirement lies. Once the subspace is found, it remains to determine the optimal solution on it. This easier task is accomplished by the least square convex optimization computed through the pseudo-inverse of the sub-dictionary.

3.3. Non-sparse ECG segments

It is natural to ask what happens when the search for a sufficiently sparse solution fails, that is the goal of finding a good approximation of the segment would require a too large support. Although this should be an exceptional condition, it is useful to manage these cases by a separate procedure. More precisely, given the normalized segment y_i of s_i, if the sparsity solver produces a support with more than σ coefficients (namely, non-sparsifiable segment), we compress s_i applying a backup procedure as in the statement:

$$\xi_i \leftarrow \textsc{BackUp\_Procedure}(s_i)$$

We choose to develop such a procedure by standard discrete wavelet transform (DWT), because of its time localization property that allows us to select the parts of the segment to drop or to maintain. Since the signal energy of a RR-segment is concentrated near the two R-peaks, i.e. at the segment extrema in our normalization, some central detail coefficients can be dropped. In particular, the PRD requirement is met decomposing the segment s_i using Daubechies mother wavelet into two levels at most, and discarding the central values of the detail coefficients thus obtaining $\hat{D}_1$ or $\hat{D}_2$. Hence, for every segment either 1st level coefficients $A_1, \hat{D}_1$ or 2nd level coefficients $A_2, \hat{D}_2$ have to be coded. The coding scheme is again a quantization, with a sufficient number of bits, concatenated with a delta encoding and arithmetic compression.

Putting together all the steps, namely sparsification, least-squares optimization and backup procedure, and then iterating on sparsity k, we have the algorithm, called NSER (which stands for *Natural Sparse Encoding Representation*), sketched in Algorithm 1.

Algorithm 1: NSER

```
Input: Matrices $\Phi$ and $\Phi^\dagger$, segment $s_i$, normalized segment $y_i$, initial guess $k_0$, error $\mathrm{PRD}_{\max}$, max sparsity $\sigma$
Output: either sparse solution $\alpha_i^*$ or DWT encoding $\xi_i$
$k_i \leftarrow k_0$
$\hat{\alpha}_i \leftarrow k\text{-LiMapS}(y_i, \Phi, \Phi^\dagger, k_i)$
$\alpha_i^* \leftarrow \mathrm{LSP}(\Phi, \hat{\alpha}_i)$
if $PRD(y_i, \Phi\alpha_i^*) > PRD_{max}$ then
    while $PRD(y_i, \Phi\alpha_i) > PRD_{max}$ do
        $k_i \leftarrow k_i + 1$
        $\hat{\alpha}_i \leftarrow k\text{-LiMapS}(y_i, \Phi, \Phi^\dagger, k_i)$
        $\alpha_i^* \leftarrow \mathrm{LSP}(\Phi, \hat{\alpha}_i)$
        if $k_i > \sigma$ then
            $\xi_i \leftarrow \mathrm{BackUp\_Procedure}(s_i)$
            break
        end
    end
else
    while $PRD(y_i, \Phi\alpha_i^*) < PRD_{\max}$ do
        $k_i \leftarrow k_i - 1$
        $\alpha_i \leftarrow k\text{-LiMapS}(y_i, \Phi, \Phi^\dagger, k_i)$
        $\alpha_{OLD} \leftarrow \alpha_i^*$
        $\alpha_i^* \leftarrow \mathrm{LSP}(\Phi, \hat{\alpha}_i)$
    end
    $\alpha_i^* \leftarrow \alpha_{OLD}$
end
```

Remark: ill-conditioned dictionary and problem regularization

A key point to discuss when applying both our sparsity or linear algebra techniques is whether the problem is well-conditioned or not. In this subsectin we addresses this aspect with regard to the derivation of pseudo-inverses involved in steps (3) and (4) respectively.

In the two above computational steps (3) and (4) we are faced with the computation of $\Phi^\dagger$ and $\Phi_{S_i}^\dagger$, that are the pseudo-inverse of $\Phi \in \mathbb{R}^{n\times m}$ (with $n < m$) and $\Phi_{S_i} \in \mathbb{R}^{n\times k}$ (with $k \ll n$) respectively. Both the matrices may be affected by rank deficiency due to the zero-padding normalization provided in the dictionary construction preliminary task. As a consequence, they may have a very high condition number[2], potentially inducing arbitrarily large errors in the solution found [11]. To avoid this drawback we perform a Tikhonov regularization step: exploiting the fact that $\Phi^\dagger = \lim_{\delta\to 0} \Phi^T(\Phi\Phi^T + \delta I)^{-1}$, in order to calculate an approximate pseudo-inverse $\Phi^\dagger$ of the dictionary, say $\Phi_R^\dagger = \Phi^T(\Phi\Phi^T + \delta I)^{-1}$, we have to set a perturbation $\delta > 0$ sufficiently small (2 or 3 order of magnitude smaller with respect to Φ's entries). Performing this regularization, the error due to the approximation $\Phi_R^\dagger\Phi \approx I$ significantly decreases, hence giving sound and competitive performances (see Sec. 4).

[2] In numerical analysis the condition number of a matrix A is defined as

$$\kappa(A) = \|A\|\,\|A^\dagger\| = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

with $A^\dagger$ being the Moore-Penrose pseudo-inverse, $\sigma_{\max}$ and $\sigma_{\min}$ being the maximum and minimum singular values respectively, and $\|\cdot\|$ being a suitable matrix norm, usually the induced 2-norm. It is understood that $\kappa(A) = \infty$ if $\sigma_{\min}(A) = 0$. The condition number of a matrix A is a measure of sensitivity of the solution of the linear problem $Ax = y$ to small perturbations of the data; in fact when $\kappa(A)$ is very large the problem is said to be ill-conditioned, while it is said to be well-conditioned when $\kappa(A) \approx 1$.

As for the worst-case computational complexity analysis of the proposed algorithm, the k-LiMapS sparsity solver requires $O(m(n + \log m))$ as explained in [3]. The least squares projection requires computing the pseudo-inverse for a rank-k submatrix (with k taking the values of the loop-variable k_i in the algorithm), that is calculated with a Singular Value Decomposition in $O(n^2k)$ time [21]. The number of iterations of the while loops is bounded by the sparsity threshold σ since the loop-variable k_i is monotonic (and $k_0 \ll \sigma$ by simply assigning it a moving average value). Therefore the total computational complexity is given by a number of floating-point arithmetic operations equal to $T_{\mathrm{NSER}}(n, m, \sigma) = \sum_{k=1}^{\sigma} O(m(n + \log m) + n^2k) = O(m(n + \log m)\sigma + n^2\sigma^2)$. Notice that the backup procedure implemented with the DWT filter banks needs $O(n)$ multiplications/additions [23] since the filter length is $\ell \ll n$, and hence its time is negligible.

4. Experimental results

We have made extensive experiments of our and competitor algorithms on the records in MIT-BIH Arrhythmia Database obtained from PhysioNet [10]. This database consists of 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH Arrhythmia Laboratory. The recordings were digitized at $F_s = 360$ samples/second per channel with unipolar ADC of 11-bit resolution over a ±5 mV range (with 1024 corresponding to the baseline 0 Volt), and are labelled with numbers in the interval 100 – 234. The tests have been conducted using the software MATLAB R2013a running on top of an Intel Core i7-4770K@3.5GHz with 32GB memory and operating system Ubuntu 14.04 LTS.

4.1. Comparative results

For comparison, we experimented also the compression of the same records with the following four methods known in literature, preserving the parameter settings reported in the original papers. Briefly, [4] proposed ARLE (adaptive run length encoding), a wavelet-based compression method that encodes runs of same symbols with an adaptive number of quantization bits; in ARLE the number of decomposition levels was set to $\eta = 6$, retention tolerance is $\varepsilon = 1\%$ and quantization tolerance is $\varepsilon_q = 10\%$. Rajoub's method [22] is based on retaining coefficients given by wavelet decomposition with a required EPE (energy packing efficiency) and then compressing the coefficient significance map using a variable-length encoding scheme; the wavelet decomposition is performed up to the 5th level using BiorSpline (bior4.4); thresholds were set in order to retain a 99.9% EPE for A_5 coefficients and 97% EPE for D_5 coefficients, while thresholds for level 1 to level 4 coefficients were set to retain various EPE, from 85% to 99%; retained coefficients were stored in 7 bit signed representation. Benzid et al. [5] proposed another wavelet transform based method that uses a bisection algorithm to reach the user-specified PRD and the quantization of retained coefficients by TRE (two-role encoder); transformation was done up to level 5 with mother wave bior4.4, the tolerance for PRD loss due to coefficient thresholding was 1%, while the tolerance for PRD loss after quantization was 10%. Another algorithm we tested is SPIHT (set partitioning in hierarchical trees) [15], which consists in an encoder based on a set partitioning ordering defined on lists of significant wavelet coefficients that exploits the temporal orientation tree structure of the coefficients and self-similarity across different layers. The wavelet used in SPIHT was bior4.4 with 5 levels of decomposition.

For sake of uniformity, all methods were tested excluding the initial part of the records (used to construct the dictionaries for the k-LiMapS method), and compressing the remaining part (more than 20 min).

After the experiments, the results corresponding to PRD ranging in the interval 0.2 – 0.6 and PRDN ranging in 2 – 10 were collected. We report in Fig. 2 the average CR-vs-PRD and CR-vs-PRDN curves computed from the experiment results obtained on all 48 MIT-BIH records. More in details, given a set of PRD values regularly chosen in the mentioned interval, we compute for each record a curve linearly interpolating the obtained pairs (PRD,CR). Then the plotted curve of each method is derived by averaging over all computed curves. Notice that our proposed method achieves a CR higher than the competitors over both PRD and PRDN intervals. Moreover, this gap increases linearly as the PRD requirement gets larger; this is due to the sparsity model taking advantage of a few atoms representation when the error constraints are weakened.

As a second type of analysis, we inspect more into details the results for every record by subdividing the PRD range into the four bins 0.2 – 0.3, 0.3 – 0.4, 0.4 – 0.5, 0.5 – 0.6, and we group the results of each record according

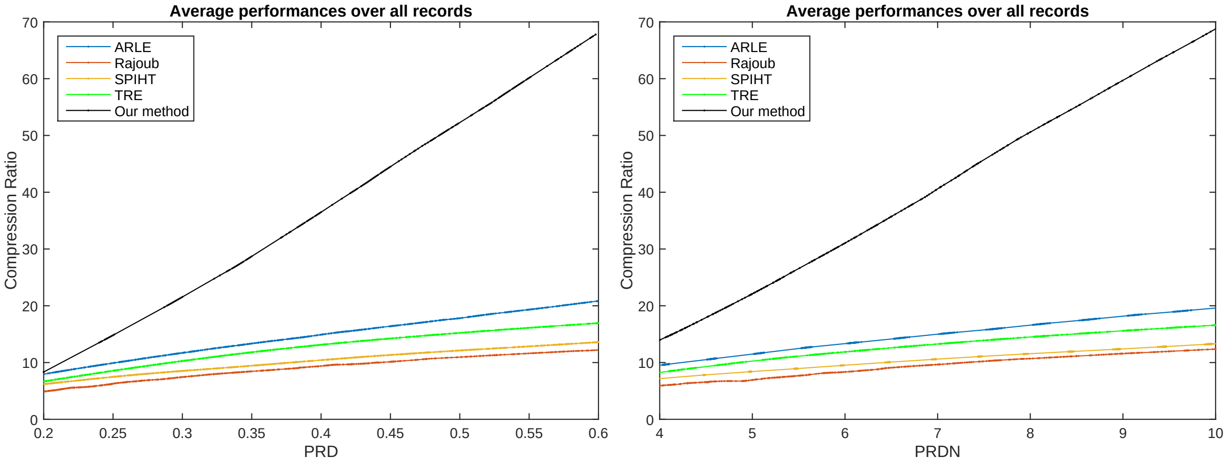

Figure 2: Average Compression Ratio among all 48 MIT-BIH Arrhythmia records versus various PRDs and PRDNs obtained by *k*-LiMapS and the 4 alternative methods. Points with different PRDs and PRDNs are extrapolated within the same record (blue: ARLE, red: Rajoub, yellow: SPIHT, green: TRE, black: *k*-LiMapS).

to this binning. Then, for each record and for each bin, we calculate the average CR. The numerical results of this analysis are plotted in Fig. 3, where points are sorted and connected for sake of readability. It is evident that, especially in the last three intervals, our method is much more effective than the 4 alternative algorithms; concerning the first interval, our method NSER performs well on half of the record set (records on the right) and outperforms the alternative algorithms on the remaining half. This behaviour of NSER with tight PRD can be explained by the fact that when the compressed part deviates largely from the atoms in the referred dictionary - as it happens with those particularly irregular records - a high number of sparsity coefficients are necessary to guarantee the required precision.

It can be interesting to view also qualitatively the reconstruction fidelity corresponding to various PRD values from 0.2 to 0.6 through our compression process. To this aim, we depict in Fig. 4 a sample of few second irregular interval extracted from record 201, that is the most compressed record by the competitor methods over all PRD values, as it can be seen from Fig. 3. Notice that, despite the high compression ratio we obtain from NSER, the error remains rather limited even increasing the PRD requirement, showing the effectiveness of our method. In particular, notice from bottom right of Fig. 4 that although the absolute error in this irregular interval is very small, namely almost below 20 corresponding to about 2%, the CR is significantly high (> 50).

4.2. Sparsity levels

In this subsection we analyze more in depth the dependence of the system performance on both the sparsity level of the solutions $\hat{\alpha}$ and the number of non-sparsifiable segments: provided a target PRD, they turn out to influence considerably the final compression ratio.

Clearly, the level of sparsity achieved by the solution of equation (2) plays a crucial role in determining the performance of the framework. The sparsity threshold $\sigma \in \mathbb{Z}$ has been introduced in order to distinguish non-sparse segments. This parameter is in fact useful for keeping under control the average complexity (i.e. number of non-zeros) of the vectors $\hat{\alpha}$ produced by the sparsity technique. Indeed, we have observed that, forcing a sparse solution for non-sparsifiable segments entails an increase of the number of non-null coefficients that have to be stored producing an inefficient encoding.

On the other hand, the Compression Ratio generally increases as the sparsity threshold σ gets larger (i.e. more segments successfully processed by the sparsity model) as it can be viewed in Figure 5, where we draw the behaviour in a range where there is a sensible variation of CR, that corresponds to low values for σ. This motivates the preference on using the sparsity model rather than the wavelet representation when convenient.

The resulting mean sparsity of coefficients $\hat{\alpha}$ obviously depends also on the PRD constraint, since stricter requirements for PRD_{max} are met only increasing the sparsity level of the representation (i.e. taking more atoms in the dictionary). For this reason we set empirically the sparsity threshold σ so that it is inversely proportional to the

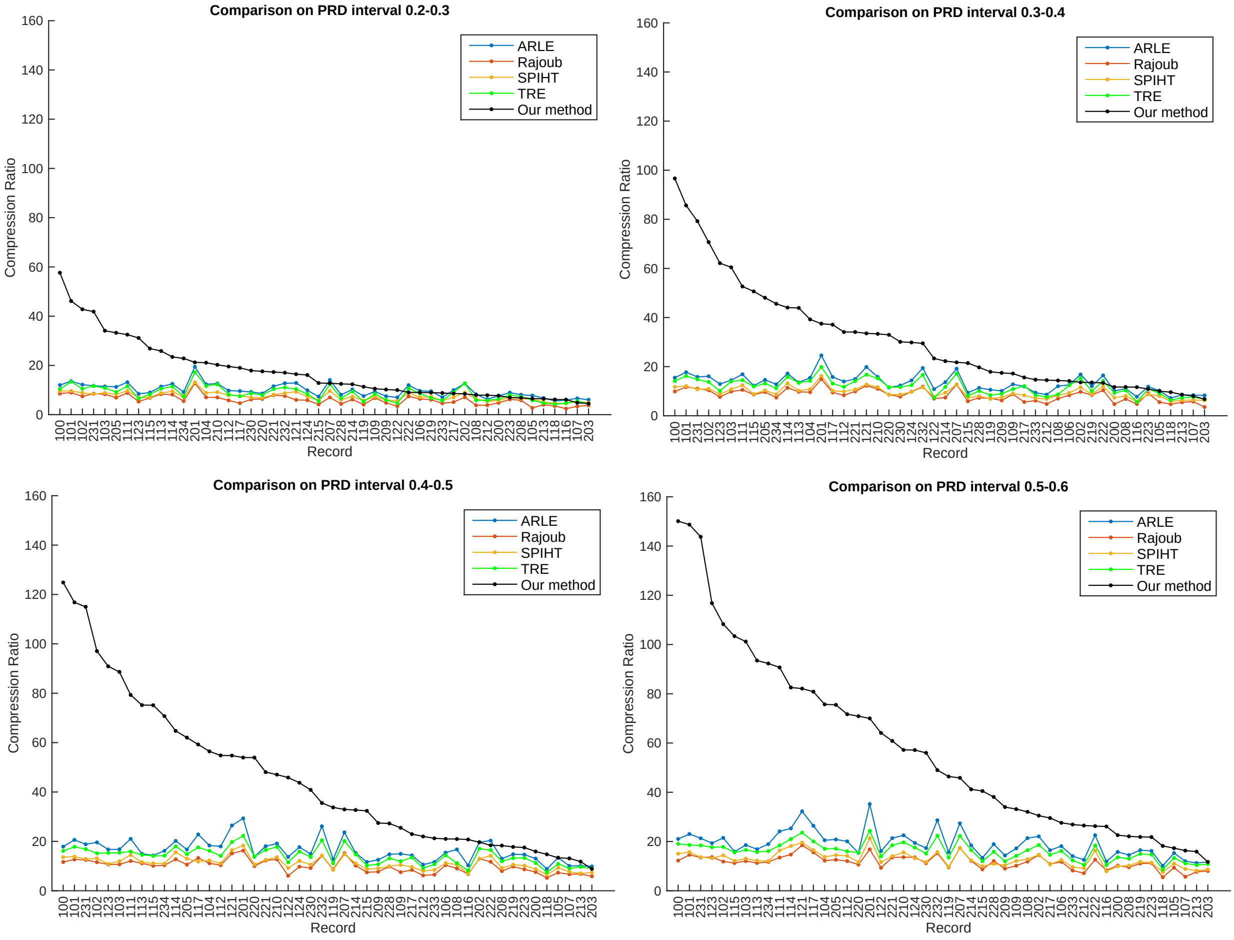

Figure 3: Comparison of the Compression Ratio obtained by all tested algorithms on the whole collection of MIT-BIH Arrhythmia Database. For every algorithm the CRs obtained by each record were grouped according to the PRD bin and then averaged. For convenience, in each graph the records (abscissae) were sorted based on the descending order of CR obtained by k-LiMapS; for the sake of readability the points were connected with a coloured solid line (blue: ARLE, red: Rajoub, yellow: SPIHT, green: TRE, black: our method).

$\mathrm{PRD}_{\max}$:

$$\sigma = \left\lfloor \frac{m}{2.5}(1 - \mathrm{PRD}_{\max}) \right\rfloor. \tag{5}$$

We now give some details of the compression process, reporting in Table 1 some inner statistics obtained on the sample records 100, 112, 205 and 234 setting various $\mathrm{PRD}_{\max}$.

It is noteworthy to see that, for representing a segment, a very small number of non-null coefficients (using the sparsity model) is sufficient in average, even if a segment has hundreds of samples. We also have confirmation that the compression rate is directly correlated to the average sparsity of the solutions $\hat{\alpha}_i$ obtained for the sparse approximation of a record. This is an expected behaviour since the less sparse the representation is, the more non-null coefficients have to be compressed by the arithmetic coding.

We also notice that, with the wavelet transform model the number of coefficients to be stored is an order of magnitude greater than those needed with the sparsity. It can be seen from the table that, with the sparsity threshold σ set as in eq. (5), the framework produces a quantity of non-sparsifiable segments that is negligible, since the records of MIT-BIH Arrhythmia contain approximately $1500 - 2000$ segments.

Finally, we notice that although the sparsity process meets the $\mathrm{PRD}_{\max}$ requirement for $\hat{s}_i$, $i = 1, ..., M$, this might not be true for the reconstructed segments $\tilde{s}_i$ due to the quantization of delta-encoded sparse coefficients, mean values

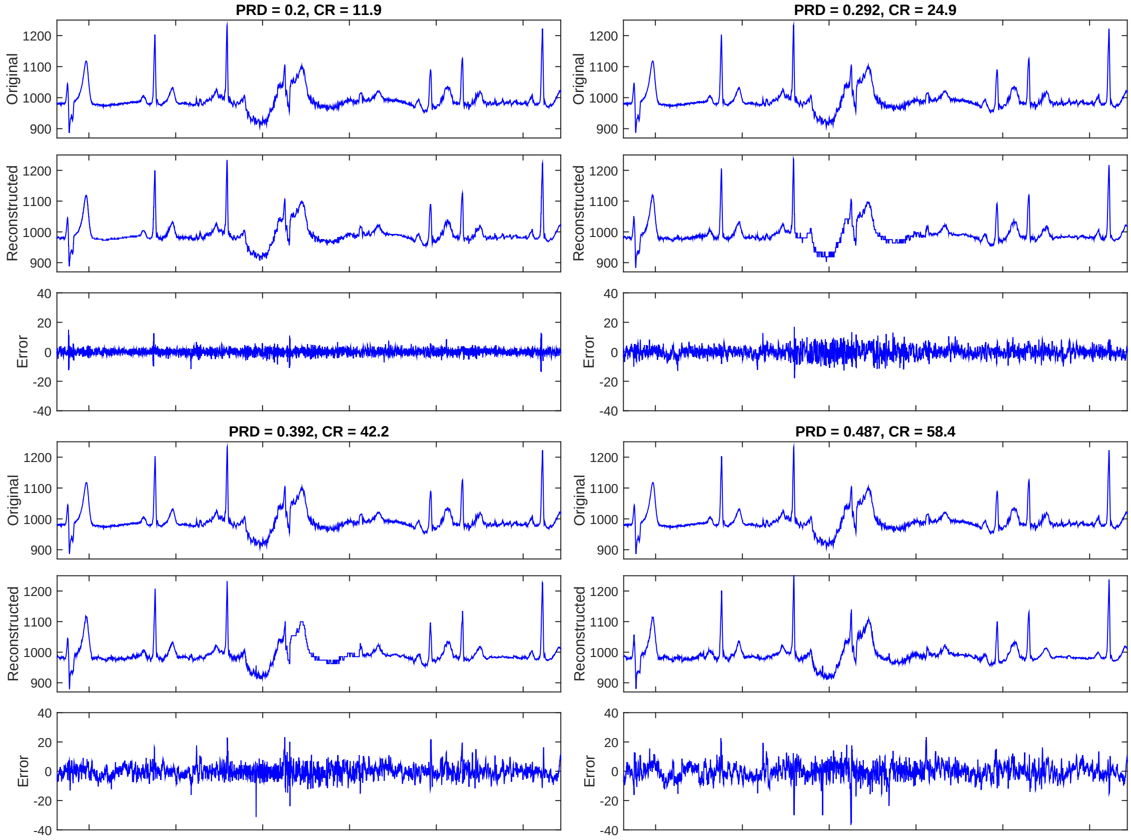

Figure 4: Qualitative view of our reconstruction fidelity corresponding to various maximum PRD values from 0.2 to 0.5, for an irregular interval extracted from sample record 201, that results to be the most compressed record by competitor methods. For each PRD we report the corresponding obtained CR on top of 3 graphs: the original signal x from the database, the version $\hat{x}$ reconstructed after compressing with our NSER method, and its absolute error $(x - \hat{x})$. Ticks on abscissa axis are marked each 500 samples, or equivalently $500/F_s \simeq 1.389$s.

and wavelet coefficients. To overcome this PRD degradation, the framework increases the quantization bits q until reaching the target PRD_{max}. To show the feasibility of this step, in Figure 6 we plot the dependency of the final PRD from different values of q for the sample records 100, 102, 112 and 119.

5. Conclusions

In this work we have proposed an online compression technique based on the sparsity solver k-LiMapS, and compared its performances in terms of CR and PRD with other wavelet-based compression algorithms known in literature: ARLE, Rajoub, SPIHT, TRE. Conducting extensive experiments on all the 48 records of MIT-BIH Arrhythmia Database, we have shown that our proposed algorithm is able to compress the tested signals achieving a ratio, that is on average 3 times higher, while respecting demanding reconstruction quality requirements. It is also evident that with the sparsity model the number of raw coefficients to be stored in memory is significantly reduced, especially when representing quasi-regular segments.

However the proposed framework is not purely sparsity-based, since in few cases of non-sparsifiable ECG segments the process resorts to a backup procedure. A natural continuation of the work should be to treat non-sparsifiable segments within the sparse framework while adding some adaptive mechanism being able to cope with irregular and unforeseen heart beats.

Considering the emergence of powerful processors for embedded systems, a possible research line should aim at developing a real-time version of the proposed technique oriented to portable devices. Another interesting direction

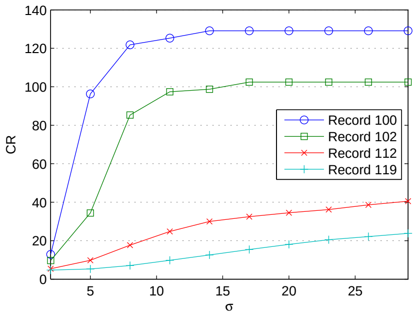

Figure 5: Curve of Compression Ratio plotted versus the sparsity threshold σ, with a reconstruction error requirement set to $\mathrm{PRD}_{\mathrm{max}} = 0.5$, for sample records 100, 102, 112 and 119. The σ has been chosen in a low range where there's a sensible variation of CR. A very high range of σ leads to a loss of compression rate due to forcing non-sparsifiable segments to be coded by the sparsity model.

Information	100	112	205	234
Record's sample frequency F_s [Hz]	360	360	360	360
Total duration in dictionary Φ [min]	7.61	6.72	6.24	6.26
Compressed signal duration [min]	22.48	23.36	23.83	23.82
Normalized segment length n [samples]	540	540	540	540
# atoms/columns in dictionary Φ	576	576	576	576
Sparsity of solution $\hat{\alpha}$ (avg±std)	5.74±5.55	17.77±19.86	18.23±18.97	4.11±8.19
Sparsity threshold σ	161.28	161.28	184.32	103.68
# non-sparsifiable segments	3	48	46	6
Avg.# stored DWT coeffs. per segment	199.33	147.58	286.65	90
# quantization bits q	6	7	9	6
Total sparsity solver time [s]	92.4	306.1	295.7	127.8
Reconstruction PRD	0.286	0.297	0.2	0.473
Reconstruction PRDN	7.225	6.975	5.094	8.201
CR	66	21.3	17.5	78.1

Table 1: Insight quantities of our proposed framework related to sparsity and coefficients to be compressed. Information on sample records 100, 112, 205 and 234 are shown.

of investigation is the learning of over-complete dictionaries containing non-patient-specific prototype signal-atoms by means of sparsity-based dictionary construction methods such as KSVD.

References

[1] Adamo, A., Grossi, G., 2011. A fixed-point iterative schema for error minimization in k-sparse decomposition. In: Proceedings of the IEEE International Symposium on Signal Processing and Information Technology (ISSPIT '11). pp. 167–172.

[2] Adamo, A., Grossi, G., Lanzarotti, R., 2012. Sparse representation based classification for face recognition by k-limaps algorithm. In: Image and Signal Processing - 5th International Conference, ICISP 2012. Vol. 7340 of Lecture Notes in Computer Science. Springer, pp. 245–252.

[3] Adamo, A., Grossi, G., Lanzarotti, R., Lin, J., 2015. ECG compression retaining the best natural basis k-coefficients via Sparse decomposition. Biomedical Signal Processing and Control 15, 11–17.

[4] Agulhari, C., Bonatti, I., Peres, P., 2013. An Adaptive Run Length Encoding method for the compression of electrocardiograms. Medical Engineering & Physics 35 (2), 145–53.

[5] Benzid, R., Marir, F., Bouguechal, N. E., 2007. Electrocardiogram compression method based on the adaptive wavelet coefficients quantization combined to a modified two-role encoder. IEEE Signal Processing Letters 14, 373–376.

[6] Chen, S. S., Donoho, D. L., Michael, Saunders, A., 1998. Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing 20, 33–61.

[7] Cover, T., Thomas, J., 2006. Elements of Information Theory, 2nd Edition. Wiley.

[8] Elad, M., 2010. Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer.

[9] Fira, M., Goras, L., Constantin, B., 2013. Reconstruction of compressed sensed ECG signals using patient specific dictionaries. In: Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS). pp. 1–4.

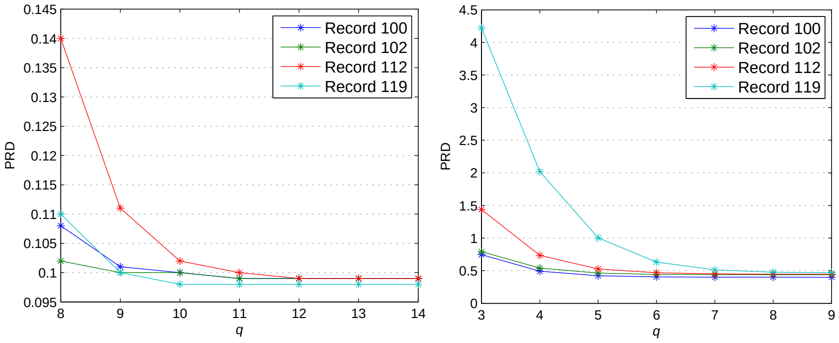

Figure 6: The effect of various quantization bits q on the final PRD (i.e. after quantization and compression) for some reconstructed signals: sample record 100, 102, 112 and 119 are shown. The PRD requirement for the sparsest approximation problem was set to $\text{PRD}_{\text{max}} = 0.1$ (on the left) and $\text{PRD}_{\text{max}} = 0.5$ (on the right). It can be easily seen that to recover the required PRD after quantization it is sufficient to choose a sufficiently large q.

[10] Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., Stanley, H. E., 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation 101 (23), e215–e220.

[11] Horn, R., Johnson, C., 2012. Matrix Analysis, 2nd Edition. Matrix Analysis. Cambridge University Press.

[12] Jalaleddine, S. M., Hutchens, C. G., Strattan, R. D., Coberly, W. A., 1990. ECG data compression techniques – a unified approach. IEEE Transactions on Biomedical Engineering 37, 329–343.

[13] Kim, B. S., Yoo, S. K., Lee, M. H., 2006. Wavelet-based low-delay ECG compression algorithm for continuous ECG transmission. IEEE Trans. on Information Technology in Biomedicine 10 (1), 77–83.

[14] Li, S., Xu, L., Wang, X., 2013. A Continuous Biomedical Signal Acquisition System Based on Compressed Sensing in Body Sensor Networks. IEEE Trans. on Industrial Informatics 9 (3), 1764–1771.

[15] Lu, Z., Kim, D. Y., Pearlman, W. A., 2000. Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm. IEEE Transactions on Biomedical Engineering 47 (7), 849–856.

[16] Mamaghanian, H., Khaled, N., Atienza, D., Vandergheynst, P., 2011. Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes. IEEE Transactions on Biomedical Engineering 58 (9), 2456–66.

[17] Manikandana, M., Soman, K., 2012. A novel method for detecting R-peaks in electrocardiogram (ECG) signal. Biomedical Signal Processing and Control 7 (2), 118–28.

[18] Natarajan, B., 1995. Sparse approximate solutions to linear systems. SIAM Journal on Computing 24 (2), 227–234.

[19] Pati, Y., Rezaiifar, R., Krishnaprasad, P., 1993. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers. IEEE Comput. Soc. Press, pp. 40–44.

[20] Polania, L., Barner, K., 2013. Multi scale dictionary learning for compressive sensing ECG. In: Proc of DSP/SPE. pp. 36–41.

[21] Quarteroni, A., Sacco, R., Saleri, F., 2007. Numerical Mathematics. Springer, Berlin, Heidelberg, Germany.

[22] Rajoub, B. A., 2002. An efficient coding algorithm for the compression of ECG signals using the wavelet transform. IEEE Transactions on Biomedical Engineering 49, 355–362.

[23] Rioul, O., Duhamel, P., March 1992. Fast algorithms for discrete and continuous wavelet transforms. IEEE Transactions on Information Theory 38 (2), 569–586.

[24] Rubinstein, R., Bruckstein, A., Elad, M., 2010. Dictionaries for sparse representation modeling. Proceedings of the IEEE 98, 243–250.

[25] Tai, S., Sun, C., Yan, W., 2005. A 2-D ECG compression method based on wavelet transform and modified SPIHT. IEEE Trans. Biomed. Eng. 52 (6), 999–1008.

[26] Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Methodological) 58.

[27] Zigel, Y., Cohen, A., Katz, A., 2000. The weighted diagnostic distortion (WDD) measure for ECG signals compression. IEEE Transactions on Biomedical Engineering, 1422–1430.