Simultaneous Greedy Analysis Pursuit for Compressive Sensing of Multi-Channel ECG Signals

Yurrit Avonds, Yipeng Liu, and Sabine Van Huffel
SCD-SISTA/iMinds Future Health Department
Department of Electrical Engineering, KU Leuven
Kasteelpark Arenberg 10, box 2446, 3001 Leuven, Belgium
email: yurrit.avonds@student.kuleuven.be; yipeng.liu@esat.kuleuven.be; sabine.vanhuffel@esat.kuleuven.be

***Abstract*—This paper addresses compressive sensing for multi-channel ECG signals. In contrast to the traditional sparse approach where the signal is decomposed into the product of a dictionary and a sparse vector, the recently developed cosparse approach exploits sparsity of the product of an analysis matrix and the original signal. We apply the cosparse approach based Greedy Analysis Pursuit (GAP) algorithm to improve the signal recovery performance of compressive sensing for ECG signals. Moreover, to efficiently recover multi-channel ECG signals, classical GAP is generalized to a multi-channel simultaneous GAP algorithm, which simultaneously reconstructs multiple signals with similar support. Numerical experiments show that the proposed method outperforms the classical methods in terms of reconstruction speed and accuracy.**

I. Introduction

Biomedical signal and image processing applications often require large amounts of data to be acquired and stored due to required level of detail . Examples include Computed Tomography and Magnetic Resonance Imaging, where a single patient scan may require more than 100 megabyte and a gigabyte of storage respectively. This paper focuses on wireless ECG monitors, where the available energy is limited and the amount of data that can be transmitted is consequently limited as well.

Since traditional wireless ECG monitors fall short in terms of energy efficiency, compression can be used to reduce the amount of transmitted data over energy-hungry wireless body sensor network links. Currently, the signal is sampled at a high rate and then compressed, which requires temporary storage of a large amount of uncompressed data before reducing it, as well as a higher sampling rate.

Compressive sensing (CS) provides a solution by compressing while sampling, avoiding the high sampling rate and storage requirements imposed by the Nyquist rate. For illustration convenience, the sampling model is formulated in the discrete setting as $\mathbf{y} = \boldsymbol{\phi}\mathbf{x}$, where $\boldsymbol{\phi} \in \mathbb{R}^{n \times N}$ and $n < N$, making measurement $\mathbf{y}$ a compressed form of signal $\mathbf{x}$.

To exploit sparsity for signal recovery, a signal $\mathbf{x}$ can be represented as $\mathbf{x} = \boldsymbol{\psi}\mathbf{s}$, i.e. a linear combination of the columns from a dictionary or basis ψ. If $\mathbf{s}$ contains a small number of nonzero elements, it is said to be a sparse vector. To estimate the sparse vector $\mathbf{s}$ from $\mathbf{y}$, $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$, we can formulate the synthesis-based ℓ_0-minimization problem:

$$\min_{\mathbf{s}} ||\mathbf{s}||_0, \quad s.\ t. \quad \mathbf{y} = \boldsymbol{\phi}\boldsymbol{\psi}\mathbf{s} \tag{1}$$

where $||\mathbf{s}||_0$ counts the number of nonzero elements in $\mathbf{s}$.

Recently, a cosparse approach was introduced, stating that an analysis matrix $\boldsymbol{\Omega} \in \mathbb{R}^{p \times N}$ can provide a sparse representation $\boldsymbol{\Omega x}$ [1]. In order to recover $\mathbf{x}$, an analysis-based ℓ_0-minimization problem can be formulated:

$$\min_{\mathbf{x}} ||\boldsymbol{\Omega}\mathbf{x}||_0, \quad s.\ t. \quad \mathbf{y} = \boldsymbol{\phi}\mathbf{x} \tag{2}$$

To solve the cosparse approach based nonconvex programming problem (2), there are two popular ways. The Analysis-by-Synthesis (ABS) method is a convex relaxation, which rewrites the analysis-based problem into the form of a synthesis-based problem [2]. The greedy algorithms contain Greedy Analysis Pursuit (GAP) which is the analysis duality for Orthogonal Matching Pursuit (OMP), and Analysis CoSaMP (A-CoSaMP) which is the analysis equivalent of Compressive Sampling Matching Pursuit (CoSaMP) [1], [3], [4].

Compared to previous works on CS for single-channel ECG, the processed data is large scale, since multiple channels are processed simultaneously. In this case, multi-channel algorithms can be used to recover the multi-channel ECG signals. They assume that channels share a similar support [5]. To the best of our knowledge, all current multi-channel methods, such as simultaneous orthogonal matching pursuit (SOMP) [6]; multiple response extension of the standard Sparse Bayesian Learning (M-SBL) [7], and Reduce MMV and Boost (ReMBO) [8], are based on a sparse representation of the synthesis form. However, our recent research shows that a cosparse approach based convex relaxation can give higher reconstruction accuracy for ECG signals than the classical sparse convex relaxation methods [9].

In this paper, the single-channel GAP algorithm is accelerated by precalculating matrices that are used in the algorithm, instead of recalculating them during each iteration and is then generalized to process multi-channel signals. The choice for a greedy algorithm is motivated primarily by its lower computational complexity. The algorithm is used for simultaneous reconstruction of multi-channel ECG signals. Numerical experiments show that the proposed algorithm achieves better reconstruction accuracy without increasing computational complexity compared to sparse methods for multi-channel ECG signal reconstruction, such as SOMMP (a faster version of Simultaneous OMP (SOMP), selecting multiple dictionary elements per iteration).

The remaining part of the paper is organized as follows. In Section II the new reconstruction algorithm is presented. In Section III the performed experiments are described. In Section IV the results of these experiments are presented. Finally, in Section V a conclusion about the results is drawn.

II. Methods

In this section, we first improve GAP by introducing precalculations of certain matrices used in the algorithm. Next, the new Simultaneous Greedy Analysis Pursuit (SGAP) method is proposed by generalization of the improved GAP.

A. GAP

While OMP searches the support of the sparse vector $\mathbf{s}$ to give a sparse estimate [10], the GAP algorithm finds the co-support (collection of zero elements) of $\mathbf{\Omega x}$. This is done by first initializing a full co-support $\hat{\Lambda}_k = \{1, 2, 3, ..., p\}$. In each iteration, the elements in the co-support that correspond to large values in the product of $\mathbf{\Omega}$ and the current estimate of $\hat{\mathbf{x}}$ are removed from the co-support estimate. Ideally, this leads to the removal of all nonzero elements from the co-support estimate [1], [3].

During the co-support update, only the index of the largest element in $\boldsymbol{\alpha} = \Omega\hat{\mathbf{x}}_{k-1}$ is selected and removed from the co-support. It is also possible to select the indices corresponding to the t largest values in $\boldsymbol{\alpha}$. In this paper, t is set to 10. The maximum number of iterations K_{max} is set as $\lfloor (p-t)/t \rfloor$.

The second stopping criterion requires the relative change in norm of the previous and current solution estimates to be larger than during the previous iteration, since this would indicate a decrease in quality of the reconstruction.

In [3], it is stated that

$$\hat{\mathbf{x}_k} = \begin{bmatrix} \boldsymbol{\phi} \\ \sqrt{\lambda}\mathbf{\Omega}_{\hat{\Lambda}_k} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} = (\boldsymbol{\phi}^T\boldsymbol{\phi} + \lambda\mathbf{\Omega}_{\hat{\Lambda}_k}^T\mathbf{\Omega}_{\hat{\Lambda}_k})^{-1}\boldsymbol{\phi}^T\mathbf{y} \quad (3)$$

with λ a small positive constant. Since neither $\boldsymbol{\phi}$, nor $\mathbf{y}$ change their value during the reconstruction process, precalculations of the result of two matrix multiplications ($\boldsymbol{\phi}^T\boldsymbol{\phi}$ and $\boldsymbol{\phi}^T\mathbf{y}$) can be done in the initialization phase of the algorithm. These precalculations result in a reduction in processing time, especially when $\boldsymbol{\phi}$ is large.

B. SGAP

The SGAP algorithm extends GAP for multi-channel measurements, based on the fact that the ECG signals in all channels share a similar co-support, since each channel in a multi-channel ECG recording can be seen as a projection of the electrical cardiac activity in the direction of the electrode associated with the channel [11]. All multi-channel signals $\mathbf{x}$ and measurements $\mathbf{y}$ are now represented by capital letters ($\mathbf{X}$, $\mathbf{Y}$) to indicate that they are matrices instead of vectors.

To generalize the GAP, several changes were made. The first difference is the addition of precalculations for speed improvement, as discussed in II-A.

Secondly, since the result of the projection $\boldsymbol{\alpha} = \Omega\hat{\mathbf{X}}$ is now a matrix instead of a vector, a different approach to find positions in the co-support that correspond to nonzero elements is required. A row-wise summation of $\boldsymbol{\alpha}$ creates a vector with large values at positions that correspond to elements that are not part of the co-support of some or all of the channels.

Finally, the second stopping criterion is adapted. Using a column-wise norm calculation, a vector with one value representing the change in norm compared to the previous solution estimate for each of the channels can be obtained. The algorithm iterates until the change in norm for any current channel becomes larger than the change in norm of that channel in the previous iteration.

Like in the GAP algorithm, 10 elements are removed from the co-support in each iteration. K_{max} is set like in the GAP algorithm as well. The complete SGAP algorithm is presented as Algorithm 1.

Algorithm 1: SGAP

In: $\mathbf{Y} \in \mathbb{R}^{n\times c}$, $\boldsymbol{\phi} \in \mathbb{R}^{n\times N}$, $\mathbf{\Omega} \in \mathbb{R}^{p\times N}$
Out: $\hat{\mathbf{X}} \in \mathbb{R}^{N\times c}$

Initialization ($k = 0$)

- Initialize Co-Support: $\hat{\Lambda}_k = \{1, 2, 3, ..., p\}$
- Precalculations
  - $\boldsymbol{\phi}_\phi = \boldsymbol{\phi}^T\boldsymbol{\phi}$
  - $\boldsymbol{\phi}_Y = \boldsymbol{\phi}^T\mathbf{Y}$
- Initialize Solution: $\hat{\mathbf{X}}_k = (\boldsymbol{\phi}_\phi + \lambda\mathbf{\Omega}^T\mathbf{\Omega})^{-1}\boldsymbol{\phi}_Y$

Iterations $k \rightarrow k+1$

- Project: $\boldsymbol{\alpha} = \Omega\hat{\mathbf{x}}_{k-1}$
- Row-Wise Summation: $\boldsymbol{\alpha}_R = \boldsymbol{\alpha}.\begin{bmatrix}1\\1\\.\\.\\.\\1\end{bmatrix}$
- Update Co-Support: $\hat{\Lambda}_k = \hat{\Lambda}_{k-1} \backslash \{\underset{i\in\hat{\Lambda}_{k-1}}{argmax}\ |\boldsymbol{\alpha}_{Ri}|\}$
- Update Solution: $\hat{\mathbf{x}}_k = (\boldsymbol{\phi}_\phi + \lambda\mathbf{\Omega}_{\hat{\Lambda}_k}^T\mathbf{\Omega}_{\hat{\Lambda}_k})^{-1}\boldsymbol{\phi}_Y$, where $\mathbf{\Omega}_{\hat{\Lambda}_k}$ is $\mathbf{\Omega}$ with the rows with row numbers not in $\hat{\Lambda}_k$ set to 0

Stopping Criteria

- $k < K_{max}$
- $\mathbf{r}_k(i) > \mathbf{r}_{k-1}(i)$ for $i \in \{1, ..., c\}$

$$\mathbf{r}_k = \begin{bmatrix} 1 - \frac{||\hat{\mathbf{x}}_k||_{2,col}(1)}{||\hat{\mathbf{x}}_{k-1}||_{2,col}(1)} \\ 1 - \frac{||\hat{\mathbf{x}}_k||_{2,col}(2)}{||\hat{\mathbf{x}}_{k-1}||_{2,col}(2)} \\ \cdots \\ 1 - \frac{||\hat{\mathbf{x}}_k||_{2,col}(c)}{||\hat{\mathbf{x}}_{k-1}||_{2,col}(c)} \end{bmatrix}$$

$$||\hat{\mathbf{x}}_k||_{2,col} = \sqrt{\begin{bmatrix}1\\1\\\cdots\\1\end{bmatrix}^T \begin{bmatrix} \hat{\mathbf{x}}_{k,11}^2 & \hat{\mathbf{x}}_{k,12}^2 & \cdots & \hat{\mathbf{x}}_{k,1c}^2 \\ \hat{\mathbf{x}}_{k,21}^2 & \hat{\mathbf{x}}_{k,22}^2 & \cdots & \hat{\mathbf{x}}_{k,2c}^2 \\ \vdots & & \ddots & \vdots \\ \hat{\mathbf{x}}_{k,N1}^2 & \hat{\mathbf{x}}_{k,N2}^2 & \cdots & \hat{\mathbf{x}}_{k,Nc}^2 \end{bmatrix}}$$

III. Numerical Experiments

To quantify SGAP performance, the MIT-BIH Arrhythmia Database[**?**], a commonly used ECG database, is used. It consists of 2-channel ECG recordings from 48 patients, sampled at 360 Hz. A total of 943 2-second segments of the first 23 patients in the database (the first 41 segments of each patient) are compressed and subsequently reconstructed. Segments of

only 2 seconds are used to keep the processing time per segment relatively low.

In order to compare the performance of SGAP to a corresponding sparse algorithm, SOMMP with a Daubechies wavelet dictionary is used. Moreover, OMMP with a Daubechies wavelet dictionary and GAP are used to reconstruct each channel separately to demonstrate the performance improvement of multi-channel algorithms over single-channel algorithms. Both SOMMP and OMMP select 4 support elements per iteration, since it was empirically found that this renders the most accurate reconstruction results for the 2-second segments used in our experiments.

The analysis matrix used in these experiments is a second order derivative matrix, obtained by multiplying 2 first order derivative matrices. This type of matrix renders faster and more qualitative results than a first order derivative matrix.

$$\mathbf{\Omega}_2 = \mathbf{\Omega}_1^2 = \begin{bmatrix} 1 & -1 & 0 & 0 & \dots & 0 \\ 0 & 1 & -1 & 0 & \dots & 0 \\ \vdots & & \ddots & & & \vdots \\ \vdots & & & \ddots & & 0 \\ \vdots & & & & 1 & -1 \\ 0 & \dots & \dots & 0 & 0 & 1 \end{bmatrix}^2 \tag{4}$$

The amount of compression applied to a signal is expressed as the compression ratio

$$CR = \frac{n}{N} \tag{5}$$

Eight different compression ratios, ranging from 0.9 to 0.2 in steps of 0.1, are used in the experiments.

In order to quantify the reconstruction accuracy, the percentage root-mean-square difference (PRD) and normalized mean absolute error are used [14], [15].

$$PRD = \frac{||x - \hat{x}||_2}{||x||_2} \tag{6}$$

$$NMAE = \frac{\frac{1}{N}\sum_i |x_i - \hat{x}_i|}{x_{max} - x_{min}} \tag{7}$$

where x_{max} and x_{min} are the largest and smallest value in the original signal, respectively.

To quantify the speed of the algorithm, the processing time and the number of iterations until convergence are measured. For OMMP and GAP, these are calculated as the respective sums of the processing times and iterations from the separate channel reconstructions.

All of these experiments were performed in MATLAB®2013a, on a quadcore 3.10GHz Intel®Core™i5-3450 system with 8GB of memory, running the CentOS 6.4 operating system with Linux kernel version 2.6.32.

IV. Results

All of the results are presented as boxplots that visualize five characteristic values and a series of outliers for each CR value. These are, in increasing order of magnitude: the lower whisker (LOW), the 25th percentile ($P25$), the median (MED), the 75th percentile ($P75$) and the higher whisker ($HIGH$), where $LOW = P25 - w.(P75 - P25)$ and $HIGH = P75 + w.(P75 - P25)$, with $w = 1.5$. The outliers (values outside the $[LOW, HIGH]$ range) are shown as crosses.

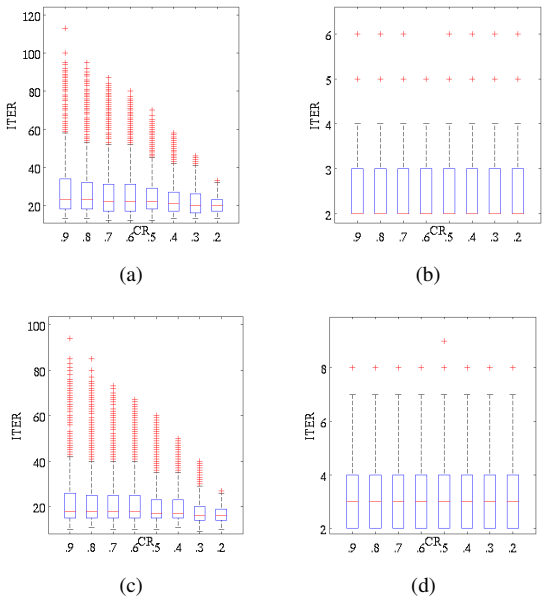


Fig. 1: Number of iterations of (a) SOMMP, (b) SGAP, (c) OMMP and (d) GAP as a function of CR for both channels.

A. Reconstruction Speed

The amount of iterations and processing times of all algorithms are presented in Fig. 1 and 2 respectively. The (S)GAP algorithm requires only a small amount of iterations to converge compared to (S)OMMP. This is because the estimate of the co-support in (S)GAP is merely used as a condition for the inversion problem and therefor it is not necessary for the co-support estimate to be very close to the actual co-support. The support estimate in (S)OMMP, on the other hand, actually determines which elements in the dictionary are used for the signal estimate and therefor needs to be close to the actual support for the algorithm to converge to a solution.

Note that during each iteration 4 elements are added to the support in (S)OMMP, while in (S)GAP 10 elements are removed from the co-support. These numbers should actually be equal to make a comparison between the sparse and cosparse algorithms in terms of the number of iterations, though in this paper a comparison at optimal reconstruction accuracy for each algorithm is desired.

Despite this large difference in the number of iterations, SOMMP processing times are only slightly lower than those of SGAP, though SGAP times vary less than those of SOMMP.

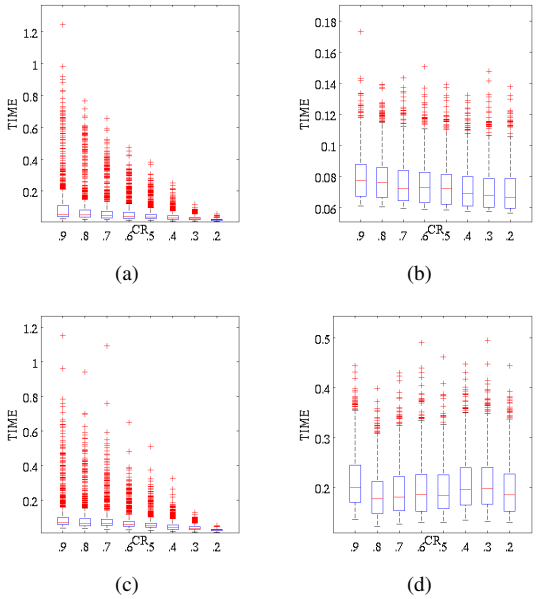


Fig. 2: Processing times of (a) SOMMP, (b) SGAP, (c) OMMP and (d) GAP as a function of CR for both channels.

This is because SOMMP solves a smaller inversion problem in each iteration, since it slowly builds a support starting from an empty one and only estimates coefficients corresponding to the support. SGAP, on the other hand, removes elements from an initally full co-support and uses the knowledge of this co-support to solve a large inversion problem to find the best possible estimate of the original signal given this knowledge.

With decreasing CR, the total number of iterations of SOMMP decreases due to the fact that its stopping criterion depends on the norm of the residue, which reduces with each iteration. This residue, which is based on the original measurement, will be smaller to begin with and reduces to a sufficiently small value faster at higher CR. In SGAP, the number of iterations remains similar for each CR value.

The total processing time required to process both channels by GAP or OMMP is significantly higher than the time required by SGAP or SOMMP to process both channels at once. This indicates that it is indeed beneficial in terms of processing time to use a multi-channel algorithm to process multiple channels at once instead of using a single-channel algorithm to process each channel separately. The reduction in time when using SGAP instead of GAP is also larger than the reduction obtained by using SOMMP instead of OMMP.

B. Reconstruction Accuracy

PRD and NMAE values of all algorithms are presented for both channels in Fig. 3 and 4 respectively. The PRD and NMAE values of the cosparse algorithms are much smaller than those of the sparse algorithms at all CR values. As expected, the quality decreases with increasing CR values for all algorithms. The sparse algorithms also show more outliers compared to their cosparse counterparts. Both multi-channel algorithms show very little difference in reconstruction quality, compared to their single-channel equivalents.

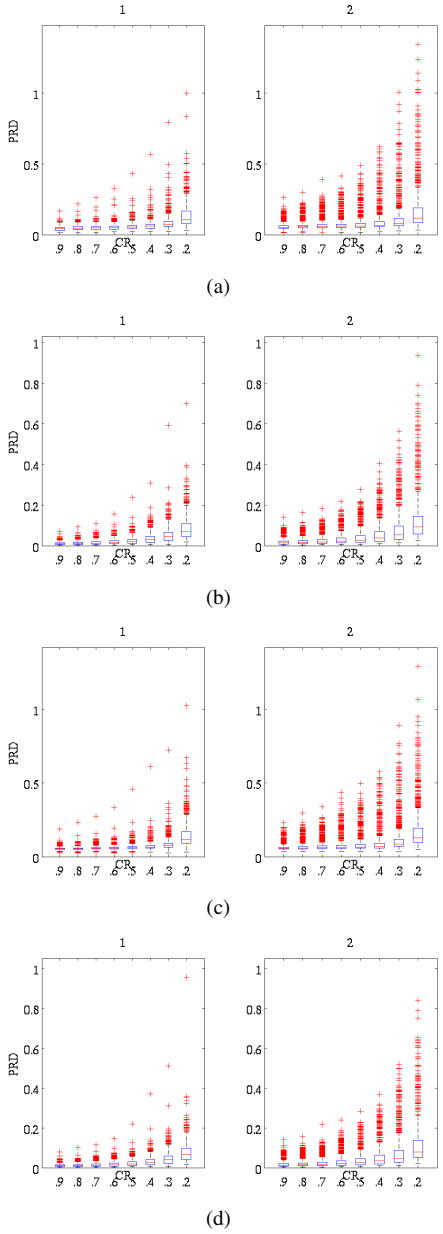


Fig. 3: PRD values of (a) SOMMP, (b) SGAP, (c) OMMP and (d) GAP as a function of CR for both channels.

These results indicate that multi-channel algorithms could be an interesting alternative to their single-channel versions, as long as the (co-)support of the different channels is similar. They also indicate that using cosparse algorithms may in fact be advantageous to the use of their sparse counterparts.

V. Conclusion

In this paper, a cosparse algorithm for simultaneous reconstruction of multi-channel ECG from compressed measurements was presented. It shows that the proposed algorithm achieves a processing time reduction, compared to the equivalent single channel algorithm. Remarkably, this does not lead to a decrease in reconstruction accuracy. Despite the

fact that processing time of SGAP is slightly higher than that of SOMMP, SGAP processing time varies less. This shows that MC algorithms can be used for qualitative, fast reconstruction of multiple signals with similar (co-)support. SGAP also renders better reconstruction quality than SOMMP, which indicates that the cosparse approach may be preferable to the classical sparse approach. Apart from the development of this algorithm, precalculations of matrices were introduced to decrease the processing time of GAP and SGAP.

ACKNOWLEDGMENT

This work was supported by Research Council KUL: GOA MaNet, PFV/10/002 (OPTEC), IDO 08/013 Autism, several PhD/postdoc and fellow grants; Flemish Government: FWO: PhD/postdoc grants, projects: G.0427.10N (Integrated EEG-fMRI), G.0108.11 (Compressed Sensing) G.0869.12N (Tumor imaging) G.0A5513N (Deep brain stimulation), I-WT: TBM070713-Accelero, TBM080658-MRI (EEG-fMRI), TBM110697-NeoGuard, PhD Grants, iMinds 2013, Flanders Care: Demonstratieproject Tele-Rehab III (2012-2014); Belgian Federal Science Policy Office: IUAP P719/ (DYSCO, 'Dynamical systems, control and optimization', 2012-2017); ESA AO-PGPF-01, PRODEX (CardioControl) C4000103224; EU: RECAP 209G within INTERREG IVB NWE programme, EU HIP Trial FP7-HEALTH/ 2007-2013 (n 260777), EU MC ITN Transact 2012 # 316679.

REFERENCES

[1] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "Cosparse analysis modeling-uniqueness and algorithms," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5804–5807.

[2] N. Cleju, M. G. Jafari, and M. D. Plumbley, "Analysis-based sparse reconstruction with synthesis-based solvers," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Kyoto, Japan: IEEE, March 25–30, 2012, pp. 5401–5404.

[3] S. Nam, M. E. Davies, M. Elad, and R. Gribonval, "The cosparse analysis model and algorithms," *Applied and Computational Harmonic Analysis*, vol. 34, no. 1, pp. 30–56, 2013.

[4] R. Giryes, M. Elad, *et al.*, "Cosamp and sp for the cosparse analysis model," in *The 20th European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, August 27–31, 2012.

[5] J. M. Kim, O. K. Lee, and J. C. Ye, "Compressive music: revisiting the link between compressive sensing and array signal processing," *Information Theory, IEEE Transactions on*, vol. 58, no. 1, pp. 278–301, 2012.

[6] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. part i: Greedy pursuit," *Signal Processing*, vol. 86, no. 3, pp. 572–588, 2006.

[7] D. P. Wipf, "Bayesian methods for finding sparse representations," Ph.D. dissertation, UC San Diego, 2006.

[8] M. Mishali and Y. C. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *Signal Processing, IEEE Transactions on*, vol. 56, no. 10, pp. 4692–4702, 2008.

[9] Y. Liu, M. De Vos, I. Gligorijevic, V. Matic, Y. Li, and S. Van Huffel, "Multi-structural signal recovery for biomedical compressive sensing," *IEEE Transactions on Biomedical Engineering*, 2013, revised.

[10] J. Tropp and S. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[11] R. Sameni, M. Shamsollahi, and C. Jutten, "Multi-channel electrocardiogram denoising using a bayesian filtering framework," in *Computers in Cardiology*. Valencia, Spain: IEEE, September 17–20, 2006, pp. 185–188.

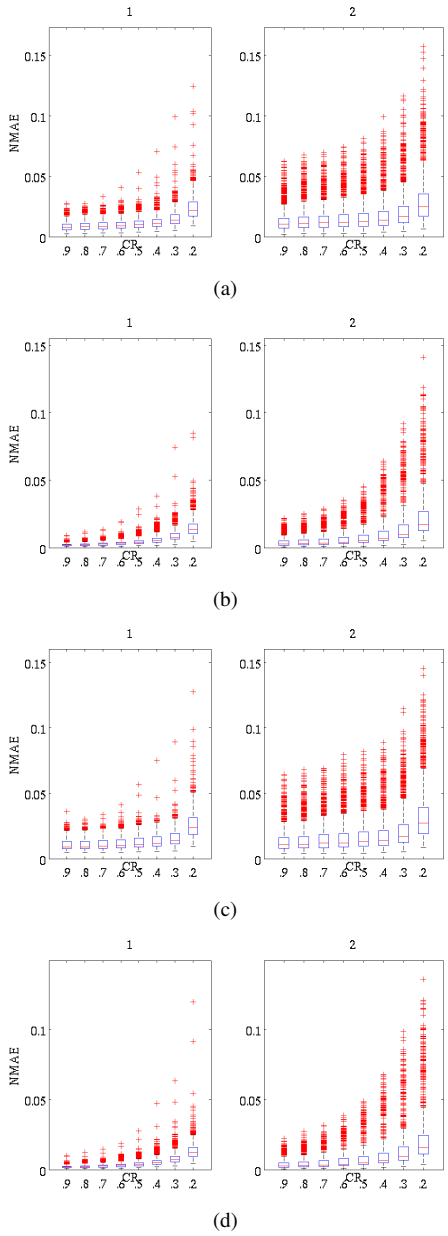


Fig. 4: NMAE values of (a) SOMMP, (b) SGAP, (c) OMMP and (d) GAP as a function of CR for both channels.

[12] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *Engineering in Medicine and Biology Magazine, IEEE*, vol. 20, no. 3, pp. 45–50, 2001.

[13] A. L. Goldberger, L. A. N. 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 (June 13).

[14] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.

[15] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ecg compression on wireless body sensor nodes," *Biomedical Engineering, IEEE Transactions on*, vol. 58, no. 9, pp. 2456–2466, 2011.