IOP Publishing | Institute of Physics and Engineering in Medicine Physiological Measurement

Physiol. Meas. **38** (2017) 1701–1713 https://doi.org/10.1088/1361-6579/aa7623

Combining sparse coding and time-domain features for heart sound classification

Bradley M Whitaker[1], **Pradyumna B Suresha**[1], **Chengyu Liu**[2], **Gari D Clifford**[2,3] **and David V Anderson**[1]

[1] Department of Electrical and Computer Engineering, Georgia Institute of Technology, GA, United States of America

[2] Department of Biomedical Informatics, Emory University, GA, United States of America

[3] Department of Biomedical Engineering, Georgia Institute of Technology, GA, United States of America

E-mail: b.whitaker@gatech.edu

Received 30 January 2017, revised 22 May 2017
Accepted for publication 31 May 2017
Published 31 July 2017

Abstract

Objective: This paper builds upon work submitted as part of the 2016 PhysioNet/CinC Challenge, which used sparse coding as a feature extraction tool on audio PCG data for heart sound classification. *Approach*: In sparse coding, preprocessed data is decomposed into a dictionary matrix and a sparse coefficient matrix. The dictionary matrix represents statistically important features of the audio segments. The sparse coefficient matrix is a mapping that represents which features are used by each segment. Working in the sparse domain, we train support vector machines (SVMs) for each audio segment (S1, systole, S2, diastole) and the full cardiac cycle. We train a sixth SVM to combine the results from the preliminary SVMs into a single binary label for the entire PCG recording. In addition to classifying heart sounds using sparse coding, this paper presents two novel modifications. The first uses a matrix norm in the dictionary update step of sparse coding to encourage the dictionary to learn discriminating features from the abnormal heart recordings. The second combines the sparse coding features with time-domain features in the final SVM stage. *Main results*: The original algorithm submitted to the challenge achieved a cross-validated mean accuracy (MAcc) score of 0.8652 ($\mathrm{Se} = 0.8669$ and $\mathrm{Sp} = 0.8634$). After incorporating the modifications new to this paper, we report an improved cross-validated MAcc of 0.8926 ($\mathrm{Se} = 0.9007$ and $\mathrm{Sp} = 0.8845$). *Significance*: Our results show that sparse coding is an effective way to define spectral features of the cardiac cycle and its sub-cycles for the purpose of classification. In addition, we demonstrate that sparse coding can be combined with additional feature extraction methods to improve classification accuracy.

Keywords: heart sound, heart sound classification, sparse coding, dictionary learning, PhysioNet/CinC Challenge

(Some figures may appear in colour only in the online journal)

1. Introduction

The introduction to this focus issue of *Physiological Measurement* sufficiently motivates the need for accurate, automated heart sound classification tools and references the 2016 Physionet/CinC Challenge. The goal of the challenge was to maximize a classifier's mean accuracy (MAcc) score, which is the average of the classifier's sensitivity (Se) and specificity (Sp)[4]. Further details can be found at https://physionet.org/challenge/2016/.

The database used in conjunction with the challenge is described in detail in Liu *et al* (2016). As a summary, the database consists of 3153 audio PCG recordings ranging from 6 to 120 s. The signals were compiled from six different databases (A)–(F) and represent recordings from healthy patients as well as patients with clinically-diagnosed cardiac diseases. The challenge organizers retained a separate test set with signals from four of the six databases as well as another two independent databases (B)–(E), (G) and (I). None of the patients represented in the hidden test set have recordings in the training set.

The material presented in this paper expands on the work submitted to the 2016 Challenge by authors Whitaker and Anderson (2016). The novelty of our solution relates to using sparse coding as a tool for performing unsupervised feature extraction. Using sparse coding in image and audio classification tasks is an active research area (Mairal *et al* (2008), Wright *et al* (2009), Elad *et al* (2010), Kavukcuoglu *et al* (2010), Wright *et al* (2010), Charles *et al* (2011), Lee *et al* (2013), Huang and Aviyente (2006)). Our own previous work has found success in sparse-domain classification tasks using support vector machines (SVMs) (Whitaker *et al* 2014).

The contributions of this paper beyond those already presented at the 2016 Computing in Cardiology Conference fall into two main categories. First, the dictionary learning process is modified by incorporating a matrix norm in the dictionary update step (Whitaker and Anderson 2015). Second, the classifier is modified to accept time-domain features in addition to the sparse coding features. The improvements in heart sound classification resulting from these changes demonstrate that sparse coding features can be combined with other application-specific features to generate a more accurate classifier.

2. Method

2.1. Audio preprocessing

Prior to extracting features and learning a classifier, we preprocessed the audio data by segmenting the heart sounds and converting the data into the frequency domain. Figure 1 offers a visual representation of the preprocessing steps.

Segmenting a PCG into periods is fundamental in the automated analysis of heard sounds (Schmidt *et al* 2010). As the first step of our algorithm we utilized Springer's state-of-the-art segmentation code, which was provided by the Challenge (Springer *et al* 2016), to separate

[4] Se = TP/(TP + FN), Sp = TN/(TN + FP), MAcc = (Se + Sp)/2. TP, TN, FP, and FN are the number of classified true positives, true negatives, false positives, and false negatives, respectively

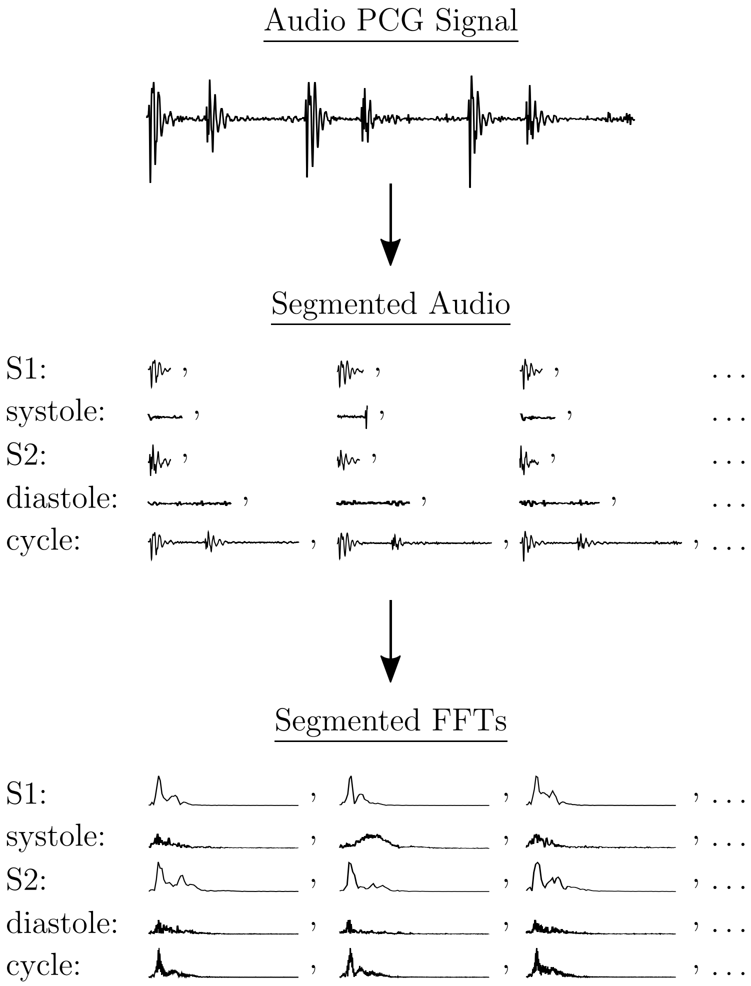

Figure 1. Visual representation of preprocessing applied to one PCG file. The preprocessing converts an audio file in the time domain into five arrays of frequency information, grouped by segmented heart sounds.

each audio file into five arrays of smaller audio segments. The first array contained a list of all S1 sounds present in the audio. The second, third, and fourth arrays contained all of the systole, S2, and diastole portions of the segmented PCG, respectively. The fifth array contained copies of the full heart cycles, starting with S1.

The next step in preprocessing the audio was to convert each sound segment from the time domain to the frequency domain with an N-point FFT. The value of N was determined by looking at the maximum lengths of the segmented heart sounds. The implementation of Springer's segmentation algorithm that we used in our implementation quantizes the sound segments' lengths to certain values. With high probability, using a random subset would produce the same maximum lengths. We selected N to be 364, 1024, 324, 2048, and 3760 for S1, systole, S2, diastole, and the full cycle, respectively. At the 2 kHz sampling frequency of the provided PCGs, these N-values correspond to 182 ms, 512 ms, 162 ms, 1.024 s, and 1.88 s, respectively.

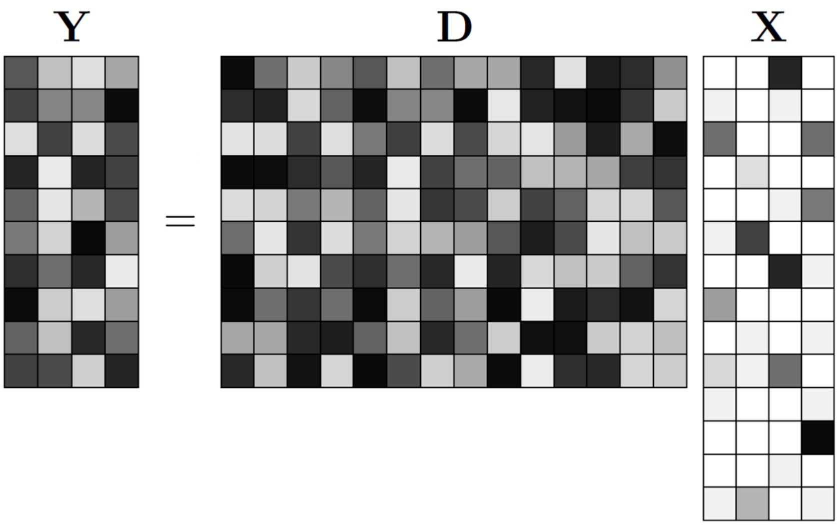

Figure 2. Visual representation of sparse coding. The data matrix (left) is factored into the product of a dictionary matrix (center) and a sparse coefficient matrix (right). Each column of the dictionary matrix can be thought of as a commonly occuring feature present in the data. The sparse coefficient matrix is a map that dictates which features are present in a given data sample.

Since the segmented audio files are purely real, the FFT is also symmetric, so we ignore the negative frequency portion of the spectrum in order to reduce computational complexity. As the spectrum magnitude contains the audible information available from the single-channel signal, we also ignore the phase to reduce computational requirements.

2.2. Sparse coding as unsupervised feature extraction

After preprocessing the PCG data, we randomly selected 1000 of the 3153 provided files from which to learn features. We used these training files to create five data matrices. The columns of the first data matrix were the preprocessed S1 segments. Likewise, the systole, S2, diastole, and full-cycle segments made up the columns for the other data matrices. We then applied sparse coding on these data matrices as a form of unsupervised feature extraction.

The goal of sparse coding is to decompose a data matrix ($\mathbf{Y}$) into the product of a dictionary matrix ($\mathbf{D}$) and a sparse coefficient matrix ($\mathbf{X}$):

$$\mathbf{Y} = \mathbf{D}\mathbf{X}. \tag{1}$$

Each column of $\mathbf{Y}$ represents a data sample, which in our case is the N-point FFT of a single subsegment of PCG audio. The dictionary matrix, $\mathbf{D}$, can be thought of as a set of commonly occurring features learned from the training data. Figure 2 gives a visual representation of sparse coding.

The intuition behind using sparse coding as a feature extraction tool is that each column of the learned coefficient matrix defines how much of each dictionary element (feature) is needed to reconstruct the respective column of the data matrix. Because the coefficient vectors are constrained to be sparse, most coefficients will be zero. This is beneficial because sparsity reduces the VC-dimension of the data, which is a measure of how difficult it is to create a good classifier (Vapnik and Vapnik 1998, Neylon 2006). Ideally in our application, the trained dictionary will have some elements that correspond to spectral patterns present in normal heart sounds. Likewise, other elements will hopefully correspond to spectral patterns associated with abnormal heart sounds.

Mathematically, performing the matrix decomposition represented in (1) and figure 2 corresponds to solving the following minimization problem (Olshausen and Field 1997, Tosic and Frossard 2011):

$$\min_{\mathbf{D}\in\mathcal{C},\{\boldsymbol{x}_{\mathrm{m}}\}} \quad \frac{1}{\mathrm{M}}\sum_{\mathrm{m}=1}^{\mathrm{M}}\frac{1}{2}\left\|\boldsymbol{y}_{\mathrm{m}}-\mathbf{D}\boldsymbol{x}_{\mathrm{m}}\right\|_2^2+\lambda\left\|\boldsymbol{x}_{\mathrm{m}}\right\|_0. \tag{2}$$

In this equation, $\boldsymbol{y}_{\mathrm{m}}$ corresponds to the m^{th} column of $\mathbf{Y}$ and $\boldsymbol{x}_{\mathrm{m}}$ corresponds to the m^{th} column of $\mathbf{X}$. The constant M represents the number of data samples that are available (number of columns of $\mathbf{Y}$). The dictionary matrix and the sparse coefficient vectors are learned simultaneously. The dictionary is constrained to $\mathcal{C}$, the set of matrices whose columns have ℓ_2-norm less than one. This prevents the dictionary from growing arbitrarily large, which would remove the effect of the ℓ_0 term in the objective function. The λ term is a fidelity-sparsity tradeoff parameter.

Unfortunately, the minimization program in equation (2) is a non-convex, NP-hard problem (Tillmann 2015). However, there are ways to approximate it and come up with workable solutions. One such method is to relax the ℓ_0-'norm' to the ℓ_1-norm and alternate solving for $\mathbf{D}$ and $\mathbf{X}$ while keeping the other constant. These relaxations result in the alternating minimization algorithm, outlined in algorithm 1 (Olshausen and Field 1997, Mairal *et al* 2014).

Algorithm 1. Alternating minimization.

Require: Signals $\{\boldsymbol{y}_{\mathrm{m}}\in\mathbb{R}^{\mathrm{N}}\}_{\mathrm{m}=1,\ldots,\mathrm{M}}$, initial dictionary $\mathbf{D}_0\in\mathcal{C}$, regularization term λ, number of iterations K

1: Initialize $\mathbf{D}\leftarrow\mathbf{D}_0$
2: **for** $\mathrm{k}=1,\ldots,\mathrm{K}$ **do**
3: **for** several $\mathrm{m}\in\{1,\ldots,\mathrm{M}\}$ (*in parallel*) **do**
4: Calculate coefficient vectors:
5: $\boldsymbol{x}_{\mathrm{m}}=\arg\min_{\boldsymbol{x}}\frac{1}{2}\left\|\boldsymbol{y}_{\mathrm{m}}-\mathbf{D}\boldsymbol{x}\right\|_2^2+\lambda\left\|\boldsymbol{x}\right\|_1$
6: **end for**
7: Update dictionary:
8: $\mathbf{D}=\arg\min_{\mathbf{D}\in\mathcal{C}}\frac{1}{\mathrm{M}}\sum_{\mathrm{m}=1}^{\mathrm{M}}\frac{1}{2}\left\|\boldsymbol{y}_{\mathrm{m}}-\mathbf{D}\boldsymbol{x}_{\mathrm{m}}\right\|_2^2$
9: **end for**
10: **return D**

The initial dictionary used in the alternating minimization algorithm is generated randomly. The number of iterations, K, is chosen somewhat arbitrarily, but should be large enough so that additional iterations would not make significant changes to the dictionary. Line 5 of the algorithm is known as 'basis pursuit denoising' (Chen *et al* 1998). This is a well-studied problem, and we chose to solve it using the software package `l1_ls` developed by Koh *et al* (2007).

In one implementation of algorithm 1, we chose to update the dictionary (line 8) using gradient descent, following the method reported in Charles *et al* (2011). We refer to this method as 'standard sparse coding'.

In a second implementation of sparse coding, we replace the dictionary update (line 8 of algorithm 1) with a matrix norm minimization problem (Whitaker and Anderson 2015):

$$\mathbf{D} = \underset{\mathbf{D}\in\mathcal{C}}{\arg\min} \|\mathbf{Y} - \mathbf{DX}\|_1 . \tag{3}$$

The norm in (3) is the matrix 1-norm, defined as

$$\|\mathbf{A}\|_1 \equiv \max_{1\leqslant \mathrm{n} \leqslant \mathrm{N}} \sum_{\mathrm{m}=1}^{\mathrm{M}} |\mathbf{A}_{\mathrm{m,n}}|. \tag{4}$$

In other words, $\|\mathbf{A}\|_1$ returns the maximum absolute column sum of $\mathbf{A}$. Therefore, minimizing $\|\mathbf{Y} - \mathbf{DX}\|_1$ will minimize the absolute deviation of any given training vector. This will encourage the learned dictionary elements to explain every data point, regardless of how often it appears. Intuitively, this will force the dictionary to represent sounds that occur infrequently in the training data, as long as the sounds have a large enough energy. Since the training data consists mostly of healthy patients, and many different illnesses are represented in the abnormal PCGs, using this dictionary approach could result in features that better discriminate between normal and abnormal heart sounds. We refer to this second implementation as 'matrix norm sparse coding'.

Applying sparse coding on the data matrices resulted in five different dictionaries. Each dictionary represents commonly occurring spectral features present in the preprocessed S1, systole, S2, diastole, and full-cycle segments. Using these dictionaries, we computed sparse coefficient vectors for each segment of each file, which are then averaged. Thus, each PCG signal is represented by five sparse coefficient vectors. This process is outlined in figure 3

2.3. Additional time-domain features

In some experiments, we include 20 time-domain features in addition to the sparse coding features explained previously. These features are described in table 1. Ten of the features are the average and standard deviation of the durations of the four heart sounds and the full cardiac cycle (RR-interval). Six features are the average and standard deviation of ratios of heart sound durations. The remaining four features are the average and standard deviation of ratios of heart sound amplitudes. All features are calculated during the segmentation preprocessing portion of the algorithm.

Recall that the sparse coding dictionaries are learned using only frequency information from the PCG signals. As a result, the learned features may not encode any time-domain information. We show that training a classifier using both the sparse coding (frequency-domain) features and the time-domain features results in higher accuracy than using the sparse codes alone.

2.4. Classification

We used the coefficient vectors from the unused 2153 files to learn five cross-validated SVM classifiers, one for each segment type (S1, systole, S2, diastole, and full cycle). We used the `libsvm` software package to learn the SVMs (Chang and Lin 2011), and we trained the SVMs using a first-order polynomial kernel. We chose this kernel because nonlinear kernels can cause extreme overfitting when classifying in a sparse domain. We tuned the other SVM parameters using the modified cuckoo search algorithm (Walton *et al* 2011). We used the soft-margin scores from each segment-specific SVM to train a final SVM that classified the PCG file to a single binary label. Figure 4 displays a visual representation of the classification process.

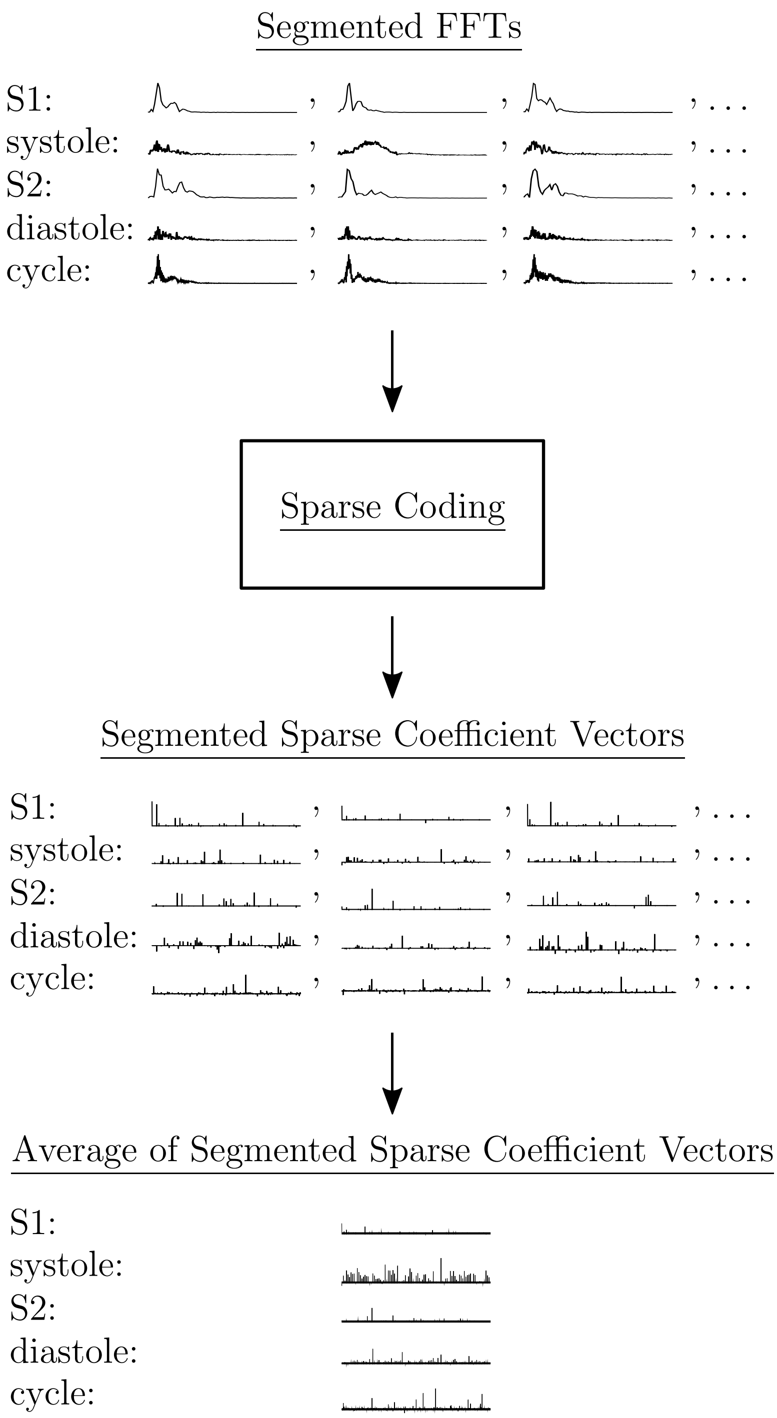

Figure 3. Visual representation of sparse coding applied to one PCG file. The sparse coding process converts each FFT from the preprocessing phase into a sparse vector representation. Each group of sparse vectors is then averaged across the PCG file, resulting in five sparse vectors—one for each heart sound segment.

Table 1. Time-domain features.

Feature	Description
μ_{RR}	Mean of RR-interval duration
σ_{RR}	Standard deviation of RR-interval duration
μ_{S1}	Mean of S1 duration
σ_{S1}	Standard deviation of S1 duration
μ_{sys}	Mean of systole duration
σ_{sys}	Standard deviation of systole duration
μ_{S2}	Mean of S2 duration
σ_{S2}	Standard deviation of S2 duration
μ_{dia}	Mean of diastole duration
σ_{dia}	Standard deviation of diastole duration
$\mu_{\mathrm{sys/RR}}$	Mean ratio of single beat's systole and RR-interval durations
$\sigma_{\mathrm{sys/RR}}$	Standard deviation ratio of single beat's systole and RR-interval durations
$\mu_{\mathrm{dia/RR}}$	Mean ratio of single beat's diastole and RR-interval durations
$\sigma_{\mathrm{dia/RR}}$	Standard deviation ratio of single beat's diastole and RR-interval durations
$\mu_{\mathrm{sys/dia}}$	Mean ratio of single beat's systole and diastole durations
$\sigma_{\mathrm{sys/dia}}$	Standard deviation ratio of single beat's systole and diastole durations
$\mu_{\mathrm{sys/S1}}$	Mean ratio of single beat's systole and S1 amplitudes
$\sigma_{\mathrm{sys/S1}}$	Standard deviation ratio of single beat's systole and S1 amplitudes
$\mu_{\mathrm{dia/S2}}$	Mean ratio of single beat's diastole and S2 amplitudes
$\sigma_{\mathrm{dia/S2}}$	Standard deviation ratio of single beat's diastole and S2 amplitudes

The classified label produced by an SVM simply reports on which side of the learned decision boundary a data point lies. The SVM soft-margin score, however, is related to how close the data point is to the decision boundary. The magnitude of the soft-margin score can be interpreted as a measure of an SVM's 'confidence' when assigning the label, and the sign of the soft-margin score corresponds to the label it would receive.

All SVMs were learned to maximize the mean accuracy score (MAcc) while including a parameter to encourage the sensitivity and specificity to be equal. Learning the five sparse coding dictionaries and the six SVMs was done offline. When we receive a new PCG file to test, we follow the same preprocessing procedure (which involves segmenting the audio and calculating the FFTs) and then learn the sparse coefficient vectors using the five dictionaries corresponding to each heart sound. After averaging the sparse vectors across the file, we generate five soft-margin scores using the segment-specific SVMs. These five scores, and optionally the 20 time-domain features, are combined into a single label using the sixth SVM, resulting in a final answer for the new file.

3. Results

As explained previously, we present four different methods of heart sound classification in our analysis. The first method is identical to the method presented at the 2016 Computing in Cardiology Conference, which used standard sparse coding features to classify the PCG audio files (Whitaker and Anderson 2016). The second method was similar, but involved the matrix norm dictionary update. The matrix norm encourages the sparse coding dictionary to learn influential features by minimizing the maximum reconstruction error. Intuitively, this would

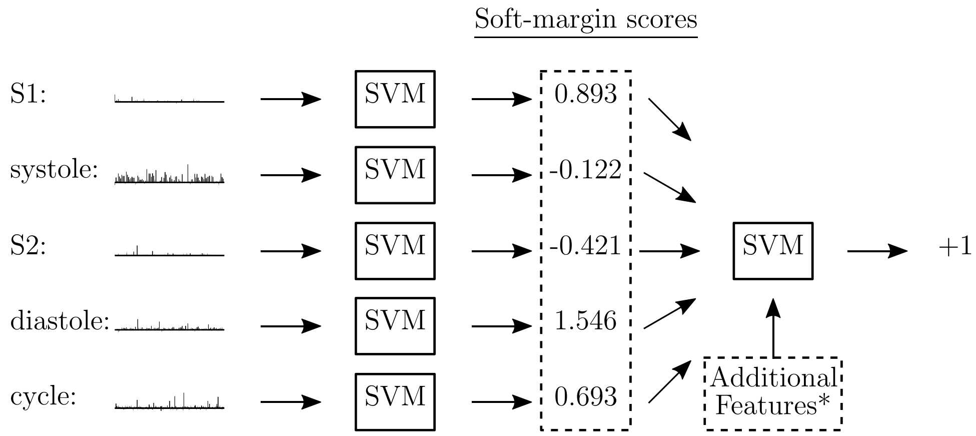

Figure 4. Visual representation of the classification process applied to one PCG file. After preprocessing and sparse coding, each file is represented by five different sparse coefficient vectors. An SVM is trained for each heart sound, and the respective sparse coefficient vector is classified with a soft margin score. These five scores are then combined using a final SVM to determine the label of a PCG file. When used, the 20 time-domain features are inputs to the final SVM as well.

Table 2. Cross-validated results.

	Se	Sp	MAcc
Standard sparse coding	0.8449	0.8537	0.8518
Matrix norm sparse coding	0.8583	0.8650	0.8615
p-value (with respect to standard sparse coding)	0.5796	0.0679	0.3124
Standard sparse coding + 20 time-domain features	0.8771	0.8771	0.8771
p-value (with respect to standard sparse coding)	0.0983	0.0747	0.0334
Matrix norm sparse coding + 20 time-domain features	0.8867	0.8816	0.8841
p-value (with respect to standard sparse coding)	0.1568	0.0113	0.0756
Zabihi *et al* (2016)	0.9423	0.8876	0.9150

allow the dictionary to learn high-energy spectral features associated with abnormal heart sounds. In standard sparse coding, that information could be lost as the average reconstruction error is minimized.

The third method uses standard sparse coding, but introduces 20 time-domain features as inputs into the final classifying SVM. Similarly, the fourth method combines matrix norm sparse coding with the time-domain features.

We repeated the procedure twice with each method in order to provide a measure of statistical reliability using the two-sample *t*-test.

Table 2 presents the ten-fold cross-validated sensitivity, specificity, and mean accuracy resulting from the two trials of each method. It also reports the *p*-values calculated from the two-sample *t*-test with respect to the standard sparse coding method. These values are calculated after classifying the 2153 files from the SVM training set and exclude the 1000 files used to learn the sparse coding dictionaries. As can be seen in the table, the matrix norm sparse coding features obtain higher sensitivity and specificity than standard sparse coding, but these improvements are not statistically significant. The table also shows that sparse coding features can be combined with other features for improved classification, as the two methods that use

Table 3. Hidden test set results.

	Se	Sp	MAcc
Standard sparse coding	0.843	0.772	0.807
Matrix norm sparse coding	0.770	0.796	0.783
Standard sparse coding + 20 time-domain features	0.801	0.806	0.803
Matrix norm sparse coding + 20 time-domain features	0.764	0.827	0.796
Potes—Hidden data Potes *et al* (2016)	0.942	0.778	0.860
Zabihi—Hidden data Zabihi *et al* (2016)	0.869	0.849	0.859

Table 4. Hidden test set results, stratified by dataset.

	Se	Sp	MAcc
Standard sparse coding	**0.843**	**0.772**	**0.807**
Dataset B	0.489	0.639	0.564
Dataset C	0.900	0.000	0.450
Dataset D	0.667	0.000	0.333
Dataset E	0.987	0.916	0.951
Dataset G	1.000	0.000	0.500
Dataset I	1.000	0.000	0.500
Matrix norm sparse coding	**0.770**	**0.796**	**0.783**
Dataset B	0.234	0.797	0.516
Dataset C	0.900	0.000	0.450
Dataset D	0.500	0.083	0.292
Dataset E	1.000	0.916	0.958
Dataset G	1.000	0.000	0.500
Dataset I	0.957	0.000	0.478
Standard sparse coding + TD features	**0.801**	**0.806**	**0.803**
Dataset B	0.532	0.589	0.560
Dataset C	0.900	0.000	0.450
Dataset D	0.583	0.083	0.333
Dataset E	0.897	0.970	0.934
Dataset G	1.000	0.000	0.500
Dataset I	0.913	0.000	0.457
Matrix norm sparse coding + TD features	**0.764**	**0.827**	**0.796**
Dataset B	0.340	0.722	0.531
Dataset C	1.000	0.000	0.500
Dataset D	0.333	0.083	0.208
Dataset E	0.962	0.973	0.967
Dataset G	1.000	0.000	0.500
Dataset I	0.870	0.000	0.435

the 20 time-domain features outperform the other two methods. When adding the additional features, the improved overall scores are significant with a p-value of 0.08.

The method that reports the best cross-validated score is the matrix norm sparse coding combined with time-domain features, resulting in Se = 0.8867, Sp = 0.8816, and MAcc = 0.8841. We also provide the CinC 2016 Challenge results from Zabihi *et al*, which represents state-of-the-art performance when measuring the cross-validated scores (Zabihi *et al* 2016, Clifford *et al* 2016). Comparing our method with Zabihi's shows that our method matches state-of-the-art performance in specificity (0.8816 versus 0.8876), but falls short with respect to sensitivity (0.8867 versus 0.9423). One possible reason for this is that we included a parameter to constrain the sensitivity and specificity to be nearly equal while training the SVMs. Removing this constraint would likely improve either sensitivity or specificity at the expense of the other metric.

While the numbers reported in table 2 show that adding the time-domain features results in an improvement in cross-validation score, they are not guaranteed to generalize to other datasets. To test the ability of our algorithms to generalize to other heart sound databases, we submitted them for evaluation on the hidden challenge test set. We present the results in table 3, along with the two best PhysioNet/CinC Challenge entries Zabihi *et al* (2016), Clifford *et al* (2016), Potes *et al* (2016).

As can be seen in the table, the algorithms did not report an improved performance on the hidden challenge data.

The full challenge results are reported in table 4. These results also report the sensitivity, specificity, and mean accuracy score achieved by our algorithms on each of the six databases represented in the hidden data. This data suggests that our classifier focuses on classifying the largest dataset (dataset E) correctly, but fails to generalize to the smaller datasets. All algorithms score between 0.934 and 0.967 on dataset E. The next highest mean score for an algorithm on a different dataset is 0.564. Several times, an algorithm classifies all samples from a given dataset as either all normal or all abnormal; this results in perfect sensitivity or specificity at the expense of the other. Moving forward, it appears that the algorithm can be improved by compensating for the number of samples in each database.

Our algorithm was implemented in MATLAB R2016a on a quad-core i7 processor clocked at 3.4 GHz with 16 GB RAM. Learning the sparse coding dictionaries and SVMs took several hours, but was done offline previous to classifying the cross-validation training samples. On average, it took our computer about 3 s to process and classify a single PCG file in the cross-validation phase. All four methods had similar computation times.

4. Conclusion

The main contribution of this paper is to introduce sparse coding as a tool for unsupervised feature extraction in heart sound classification. This paper is also the first to use matrix norm sparse coding in a practical classification setting. We recognize that our algorithm does not use an exhaustive list of features. The main goal of the paper was to show that sparse coding features could be combined with other features to improve classification. For this reason, we did not use other tools that have been used in heart sound classification, such as frequency features, the presence of S3 and S4 sounds, or features related to subintervals of systole and diastole. Further work may incorporate additional features with the hope of improving the classification score or robustness. This work focuses on the versatility of sparse coding in a classification setting. We believe that sparse coding can have a positive impact if applied to other classification tasks.

We acknowledge that the current algorithm cannot be used as a diagnostic tool for cardiac diseases; it only detects abnormalities in cardiac cycles that may be of interest to a cardiologist for further evaluation.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1650044.

ORCID

Bradley M Whitaker https://orcid.org/0000-0001-8884-9743
Chengyu Liu https://orcid.org/0000-0003-1965-3020

References

Chang C-C and Lin C-J 2011 LIBSVM: a library for support vector machines *ACM Trans. Intell. Syst. Technol.* **2** 1–27

Charles A S, Olshausen B A and Rozell C J 2011 Learning sparse codes for hyperspectral imagery *IEEE J. Sel. Top. Signal Process.* **5** 963–78

Chen S S, Donoho D L and Saunders M A 1998 Atomic decomposition by basis pursuit *SIAM J. Sci. Comput.* **20** 33–61

Clifford G D, Liu C, Moody B, Springer D, Silva I, Li Q and Mark R G 2016 Classification of normal/abnormal heart sound recordings: the Physionet/Computing in Cardiology Challenge 2016 *Proc. Computing in Cardiology Conf.* pp 609–12 (http://ieeexplore.ieee.org/document/7868816/)

Elad M, Figueiredo M A T and Ma Y 2010 On the role of sparse and redundant representations in image processing *Proc. IEEE* **98** 972–82

Huang K and Aviyente S 2006 Sparse representation for signal classification *Advances in Neural Information Processing Systems* (Cambridge, MA: MIT Press) pp 609–16 (https://papers.nips.cc/paper/3130-sparse-representation-for-signal-classification)

Kavukcuoglu K, Ranzato M and LeCun Y 2010 Fast inference in sparse coding algorithms with applications to object recognition (arXiv:1010.3467)

Koh K, Kim S-J and Boyd S 2007 An interior-point method for large-scale l1-regularized logistic regression *J. Mach. Learn. Res.* **8** 1519–55 (http://www.jmlr.org/papers/v8/koh07a.html)

Lee K, Hyung Z and Nam J 2013 Acoustic scene classification using sparse feature learning and event-based pooling *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (IEEE) pp 1–4

Liu C *et al* 2016 An open access database for the evaluation of heart sound algorithms *Physiol. Meas.* **37** 2181–213

Mairal J, Bach F and Ponce J 2014 Sparse modeling for image and vision processing *Found. Trends Comput. Graph. Vis.* **8** 85–283

Mairal J, Leordeanu M, Bach F, Hebert M and Ponce J 2008 Discriminative sparse image models for class-specific edge detection and image interpretation *Computer Vision ECCV 2008* (*Lecture Notes in Computer Science*) ed D Forsyth *et al* vol 5304 (Berlin: Springer) pp 43–56

Neylon T 2006 Sparse solutions for linear prediction problems *PhD Thesis* New York University (http://www.cs.nyu.edu/faculty/shasha/papers/neylonthesis.pdf)

Olshausen B A and Field D J 1997 Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vis. Res.* **37** 3311–25

Potes C, Parvaneh S, Rahman A and Conroy B 2016 Ensemble of feature-based and deep learning-based classifiers for detection of abnormal heart sounds *Proc. Computing in Cardiology Conf.* vol 43 pp 621–4 (http://ieeexplore.ieee.org/document/7868819/)

Schmidt S E, Holst-Hansen C, Graff C, Toft E and Struijk J J 2010 Segmentation of heart sound recordings by a duration-dependent hidden Markov model *Physiol. Meas.* **31** 513

Springer D B, Tarassenko L and Clifford G D 2016 Logistic regression-hsmm-based heart sound segmentation *IEEE Trans. Biomed. Eng.* **63** 822–32

Tillmann A M 2015 On the computational intractability of exact and approximate dictionary learning *IEEE Signal Process. Lett.* **22** 45–9

Tosic I and Frossard P 2011 Dictionary learning *IEEE Signal Process. Mag.* **28** 27–38

Vapnik V N and Vapnik V 1998 *Statistical Learning Theory* vol 1 (New York: Wiley)

Walton S, Hassan O, Morgan K and Brown M R 2011 Modified cuckoo search: a new gradient free optimisation algorithm *Chaos Solitons Fractals* **44** 710–8

Whitaker B and Anderson D 2015 Learning anomalous features via sparse coding using matrix norms *IEEE Signal Processing Workshop (Salt Lake City, USA)* pp 196–201

Whitaker B and Anderson D 2016 Heart sound classification via sparse coding *Proc. Computing in Cardiology Conf.* vol 43 pp 805–8 (http://ieeexplore.ieee.org/document/7868865/)

Whitaker B M, Carroll B T and Anderson D V 2014 Sparse decomposition of audio spectrograms for automated disease detection in chickens *Proc. IEEE GlobalSIP* (IEEE) pp 1122–6

Wright J, Ma Y, Mairal J, Sapiro G, Huang T S and Yan S 2010 Sparse representation for computer vision and pattern recognition *Proc. IEEE* **98** 1031–44

Wright J, Yang A Y, Ganesh A, Sastry S S and Ma Y 2009 Robust face recognition via sparse representation *IEEE Trans. Pattern Anal. Mach. Intell.* **31** 210–27

Zabihi M, Rad A B, Kiranyaz S, Gabbouj M and Katsaggelos A K 2016 Heart sound anomaly and quality detection using ensemble of neural networks without segmentation *Proc. Computing in Cardiology Conf.* vol 43 pp 613–6