A Tensor Approach to Heart Sound Classification

Ignacio J. Diaz Bobillo
Institute for Data, Systems and Society, Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

In the context of the PhysioNet/CinC 2016 Challenge, where a relatively large, labeled data set of phonocardiograms (PCGs) was made available, this work presents a mixed approach to the problem of its binary classification. Instead of laboriously selecting a set of PCG signal features that capture the fundamental differences between healthy and unhealthy heart sounds, a rather exhaustive set of features is generated for each heart beat segment, which is then represented in a 4-way tensor. In a second stage, such tensor representation is decomposed and compressed, to determine only a few of the most discriminating parameters, which are then fed to an otherwise standard classifier. This results in an accurate, compact and fast algorithm, that can effectively classify noisy PCG signals of different duration, achieving a balanced accuracy of 91.9% in 10-fold cross-validation, and 84.54% on the Challenge hidden test data (the 4^{th} highest score).

1 Background

Heart disease continues to be the leading cause of death in most countries, and its early diagnosis and treatment is key in improving long term health outcomes. Nowadays, the medical profession enjoys a powerful set of tools for its diagnosis, including electrocardiographs, magnetic and ultrasound scanners, echocardiographs, cardiac MRIs, 3D CT scans, etc. However, most of these instruments are only found in relatively large hospitals, require trained professionals to operate them, and medical doctors to interpret the observations. Alternatively, auscultation of the heart has been a traditional aid to the early detection of heart disease, that only requires a stethoscope and a trained listener. But with the advent of such advanced tools, medical practitioners are quickly loosing their heart auscultation training and ability [1]. The notion of an electronic stethoscope that could aid the listener or even provide a diagnosis, has eluded a practical and effective answer for decades [2]. Recent advances in machine learning and the development of small computing devices, such as smartphones, promise to soon give a more definite answer to this problem. This environment has promoted a surge of research around this topic, part of which is summarized in [2, 3], and competitions such as this Challenge [3].

Here we tackle the problem of classifying PCGs as normal or abnormal, using well known tensor-based machine learning techniques that have been successfully applied to other signal classification and data compression problems [4, 5]. Feature extraction is at the heart of an accurate classification algorithm. Tensor methods provide an effective way of reducing the dimensionality of a feature space. Here we will use a tensor decomposition approach that maximized the discriminating power of the compressed data, which is a natural extension of Discriminant Analysis to higher order data arrays.

2 Tensor decomposition and dimensionality reduction

A tensor is a multi-way array of data, and is the natural generalization of a matrix when the order is higher than two. Real tensors of order N are denoted by underlined, bold, capital letters, e.g. $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$; matrices are denoted by bold, capital letters, e.g. $\mathbf{A} \in \mathbb{R}^{I \times J}$; and vectors by bold, italic, lowercase letters, e.g. $\boldsymbol{v} \in \mathbb{R}^{I}$.

A tensor's mode-n fiber is a vector obtained by fixing all indices but the nth one. A mode-n matrization of a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, is an unfolding or flattening of the tensor along its nth way, obtained by arranging its mode-n fibers as columns of a matrix. Such matrix is represented by $\mathbf{X}_{(n)}$.

The n-mode product of a tensor and a matrix is denoted by $\underline{\mathbf{Y}} = \underline{\mathbf{X}} \times_n \mathbf{A}$, and in its equivalent matrix form, $\mathbf{Y}_{(n)} = \mathbf{A}\mathbf{X}_{(n)}$. Multiplication in all possible modes of a tensor $\underline{\mathbf{G}}$ and matrices $\mathbf{A}^{(n)}$, for $n = 1, 2, ..., N$, is denoted by $\underline{\mathbf{G}} \times_{\mathbf{1}} \mathbf{A}^{(\mathbf{1})} \times_{\mathbf{2}} \mathbf{A}^{(\mathbf{2})} \cdots \times_N \mathbf{A}^{(N)}$, or in short $\underline{\mathbf{G}} \times \{\mathbf{A}\}$. Multiplication of a tensor by all but one mode is indicated by $\underline{\mathbf{G}} \times_{-n} \{\mathbf{A}\}$, and $\underline{\mathbf{G}} \times_{-(n,m)} \{\mathbf{A}\}$ when two modes, n and m, are skipped. Finally, we define the contracted product of tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ and tensor $\underline{\mathbf{Y}} \in \mathbb{R}^{J_1 \times J_2 \cdots \times J_N}$ along all modes except mode-n, as $\langle \underline{\mathbf{X}}, \underline{\mathbf{Y}} \rangle_{-n} = \mathbf{X}_{(n)} \mathbf{Y}_{(n)}^T \in \mathbb{R}^{I_n \times J_n}$.

As a natural extension to matrix factorization, tensors can be decomposed in many different ways, the simplest being as the sum of rank-one tensors (known as CP decomposition). A rank-one tensor in $\mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, is given by the outer product of N vectors in $\mathbb{R}^{I_n}$. Now, tensor rank is an elusive and complex concept, which reflects on the theory of tensor rank approximation (for a review, see

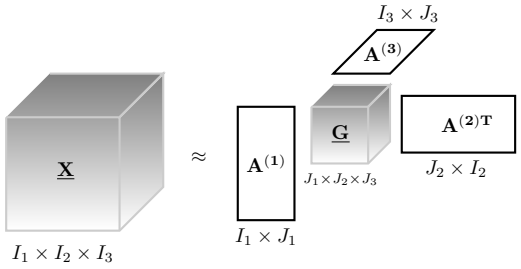

Figure 1. Tucker approximation of a 3-way tensor

[6]). A more general form, known as the Tucker decomposition, decomposes a tensor into a core tensor $\underline{\mathbf{G}}$ multiplied by matrices along each mode. When the decomposition is inexact, it is call Tucker approximation (see Figure 1 for a 3-way illustration). The Tucker decomposition is richer than the CP, in that it includes interacting elements in the core tensor. In fact, the CP decomposition is a special case where the core tensor has nonzero elements only in its superdiagonal. In short, for the general case, a Tucker approximation can be written as:

$$\underline{\mathbf{X}} \approx \underline{\mathbf{G}} \times_{\mathbf{1}} \mathbf{A^{(1)}} \times_{\mathbf{2}} \mathbf{A^{(2)}} \cdots \times_N \mathbf{A}^{(N)} = \underline{\mathbf{G}} \times \{\mathbf{A}\}$$

A special form of Tucker decomposition, known by Tucker-N, results when the $(N+1)$-th factor matrix corresponding to an $(N+1)$-order tensor, is forced to be the identity. This is illustrated in Figure 2, viewed as the simultaneous decomposition of K tensors (K being the dimension of the $(N+1)$-way), that when concatenated produces an $(N+1)$-way core tensor.

In the special but common case where factor matrices $\mathbf{A}^{(n)}$ are orthogonal, the Tucker decomposition reduces to a higher order form of principal components. For algorithmic simplicity, this is the form generally used, and factor matrices are denoted by $\mathbf{U}^{(n)}$ as is customary in such cases. Hence,

$$\underline{\mathbf{G}} \approx \underline{\mathbf{X}} \times_{\mathbf{1}} \mathbf{U^{(1)T}} \times_{\mathbf{2}} \mathbf{U^{(2)T}} \cdots \times_N \mathbf{U}^{(N)\mathbf{T}} = \underline{\mathbf{X}} \times \{\mathbf{U^T}\}$$

Given a choice of dimensions for $\underline{\mathbf{G}}$, the Tucker factor matrices can be computed to achieve a certain objective, for instance, to minimize the Frobenius norm of the approximation error (see [7]), as in data compression. However, a more effective objective for the problem at hand, is to maximize the high order generalization of the Fisher score corresponding to the core tensor. That is,

$$\varphi = \max_{\mathbf{U^{(1)}},\ldots,\mathbf{U}^{(N)}} \frac{\sum_{c=1}^{C} K_c \parallel \bar{\underline{\mathbf{G}}}_{(c)} - \bar{\bar{\underline{\mathbf{G}}}} \parallel_{\mathbf{F}}^{\mathbf{2}}}{\sum_{k=1}^{K} \parallel \underline{\mathbf{G}}^{(k)} - \bar{\underline{\mathbf{G}}}_{(c_k)} \parallel_{\mathbf{F}}^{\mathbf{2}}} \qquad (1)$$

where C is the number of classes, K_c is the number of samples in class c, $\bar{\underline{\mathbf{G}}}_{(c)}$ is the average core tensor over samples in class c, $\bar{\bar{\underline{\mathbf{G}}}}$ is the average core tensor over al samples, and $\bar{\underline{\mathbf{G}}}_{(c_k)}$ is the average core tensor for class c_k.

It can be shown (see [5]) that Eq.1 is equivalent to solving the following high-order trace-ratio optimization problem:

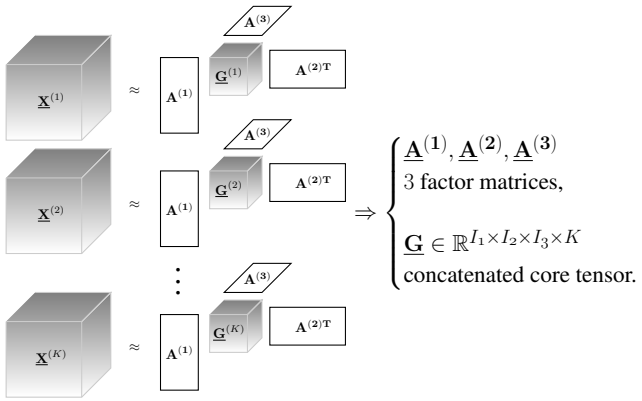

Figure 2. Simultaneous Tucker approximation

$$\varphi = \max_{\mathbf{U^{(1)}},\ldots,\mathbf{U}^{(N)}} \frac{\mathrm{tr}\left[\mathbf{U}^{(n)\mathbf{T}} \mathbf{S}_{\mathbf{b}}^{-n} \mathbf{U}^{(n)}\right]}{\mathrm{tr}\left[\mathbf{U}^{(n)\mathbf{T}} \mathbf{S}_{\mathbf{w}}^{-n} \mathbf{U}^{(n)}\right]} \qquad (2)$$

where $\mathbf{S}_{\mathbf{w}}^{-n} = \langle \tilde{\underline{\mathbf{Z}}}^{-n}, \tilde{\underline{\mathbf{Z}}}^{-n} \rangle_{-n}$ is the n-mode within-class scatter matrix, and $\mathbf{S}_{\mathbf{b}}^{-n} = \langle \breve{\underline{\mathbf{Z}}}^{-n}, \breve{\underline{\mathbf{Z}}}^{-n} \rangle_{-n}$ is the n-mode between-class scatter matrix. In this notation, $\tilde{\underline{\mathbf{Z}}}^{-n} = \tilde{\underline{\mathbf{X}}} \times_{(n,N+1)} \{\mathbf{U^T}\}$, where $\tilde{\underline{\mathbf{X}}}$ is the concatenation of $\tilde{\underline{\mathbf{X}}}^{(k)} = \underline{\mathbf{X}}^{(k)} - \bar{\underline{\mathbf{X}}}_{(c_k)}$ (centered tensors), and $\breve{\underline{\mathbf{Z}}}^{-n} = \breve{\underline{\mathbf{X}}} \times_{(n,N+1)} \{\mathbf{U^T}\}$, where $\breve{\underline{\mathbf{X}}}_{(c)} = \sqrt{\mathbf{K_c}} \left(\bar{\underline{\mathbf{X}}}_{(c)} - \bar{\bar{\underline{\mathbf{X}}}} \right)$.

Problem 2 can be solved alternating over each mode, by solving two generalized eigenvalue problems on each mode at each iteration. See [5] for algorithmic details.

3 Feature space selection and data tensor build-up

In the previous section we have established the basic mechanics for reducing the dimensionality of the data tensor, while maximizing its discriminating capacity between sample classes. In this section we define how to fill the data tensor, given a set of heart sound recordings. For this, we start with a rich time-frequency feature space, given by combining Mel Frequency Cepstral Coefficients, or MFCCs, and the Maximal Overlap Discrete Wavelet Packet Transform, or MODWPT. MFCCs were developed around the characteristics of human hearing, and are typically used in speech recognition application. Both have been used in previous work, as reported in [2, 3]. The following sequence summarizes the processing applied to each recording:

1. 1000 Hz resampling and normalization.

2. S1/Systole/S2/Diastole segmentation of each complete heart beat, by direct, unmodified use of the Springer algorithm [8].

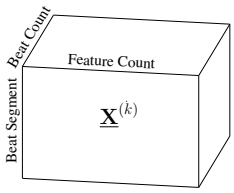

Figure 3. Tensor representation of the k-th recording

Database	Recordings	Normal	Abnormal
a	409	117	292
b	490	386	104
c	31	7	24
d	55	27	28
e	2054	1871	183
f	114	80	34
Total	**3153**	**2488**	**665**

Table 1. Summery of Training Set

3. Computation of MFCCs (using code from [9]), independently for each segmented sequence and each heartbeat, including energy, "delta" and "delta-delta" coefficients, using Hanning time windows, such that the total power of the FFT is preserved. This generates 13×3 coefficient series, of different lengths depending on the length of the input sequence segment. If the minimum length of 128 is not met, such input sequence is extended in a periodic fashion (this was found to be better than padding with zeros).
4. Computation of MODWPT independently for each segmented sequence and for each heartbeat, using a 6^{th} level decomposition of the Symlets wavelet family of order 8 (i.e., sym8). Again, if the minimum support of 64 is unmet, the input sequence is periodically extended to meet the requirement. This transformation generates 64 coefficient sequences, with lengths equal to the input sequence.
5. At this point we have expanded the feature space significantly. So next we compute some basic descriptive statistics of each coefficient sequence to carry through, namely their arithmetic mean, and their mean absolute deviation as a measure of dispersion (which was found to be more informative than the standard deviation). Hence, for each segment of each heartbeat we have $13 \times 3 \times 2$ MFCC related features, and 64 MODWPT dispersion related features (the means are near zero, so they are not included). Later experimentation showed that about half of these features have a small or null contribution to the classification performance. Hence, they could be eventually removed in a practical application.
6. These features are arranged in a 3-way tensor, as shown in Fig. 3. Then, the kth recording tensor, $\underline{\mathbf{X}}^{(k)}$, will have dimensions $142 \times 4 \times$ (# heartbeats).
7. Lastly, in order to concatenate all recording tensors into a 4-way tensor, $\underline{\mathbf{X}}$, we need to accommodate the fact that the number of heartbeats differ for each recording. Again, we are faced with the need to complete "missing" information. Since we do not want the length of the recordings to dominate the spatial structure of the data, we extend all recording features to match the longest one, by repeating the series from the start, in total or in part, as many times as necessary.

The output of this process is a dense, 4-way data tensor, of dimensions $142 \times 4 \times 172 \times$# of recordings in the data set. To this tensor we apply the methods described in Section 2 to obtain a core tensor of reduced dimensions and higher discriminating power. The choice of core tensor dimensions is done heuristically, based on classification performance, via a grid search. The forth dimension remains fixed (i.e., Tucker-3 decomposition), while the first three dimensions are varied in combinations such that the total number of elements in $\underline{\mathbf{G}}^{(K)}$ is of the same order of magnitude as the number of samples in the minority class. Then, these elements are sorted out in descending order of their Fisher scores (element-wise version of Eq. 1), and only the top F elements are fed to the classifier (around 1/10th of the total). Hence, the dimensionality reduction achieved is on the order of 1000:1. Note that the indices of the top elements need to be stored for classifying new observations.

To reflect the different amount of information in long recordings vs short (extended) ones, a weighted version of Eq. 1 was used, where all tensor averages were weighted with an ad-hoc weight defined as the number of heartbeats to the $1/3$ power (this achieved a somewhat better performance than the more principled $1/2$ power). Finally, it is worthwhile noting that no normalization is needed on the tensor. (Note: all coding was done in Matlab, using its standard toolboxes, and some functions from [10].)

4 Classification approach and results

The supplied data set is listed in Table 1 (for a detailed description see [3]). Each database corresponds to a different healthcare institution/professional who contributed the data. Early on in this investigation, it was noted that database *e* behaved differently from the rest. Not only is the largest one, but also the easiest to separate between classes (over 99% balanced accuracy). The reason for this is intriguing and worth investigating further. A review of the acquisition practices described in [3], suggests that the fact that all healthy patient recordings were done with one type of sensor, while all unhealthy ones with another type of sensor, may explain this in part. The organizers also provided signal quality labels on the recordings (good/poor) and "unsure" diagnosis label based solely on PCG listening by a group of experts. The availability of such additional information suggests that a 4-class to 3-class classifier (normal-good/normal-poor/abnormal-

Validation	10-fold CrossVal	Test Data
Bal. Accuracy	**91.9%**	**84.54%**
Sensitivity	**94.6%**	**86.39%**
Specificity	**89.3%**	**82.69%**

Table 2. KNN Performance metrics.

Training DB	Bal. Accuracy	Sensitivity	Specificity
Only *e*	98.6%	98.5%	98.7%
All but *e*	77.6%	78.5%	76.7%

Table 3. 10-fold CrossVal using different databases.

poor/abnormal-good to normal/unsure/abnormal) by posterior probability aggregation, could outperform a binary classifier. However, this was not confirmed to be the case, with the caveat that the multi-class approach was only applied in the last classification stage, and not in the tensor compression stage which remained binary (a restriction that could be removed in future work).

Several classification algorithms were tried on the resulting $F \times 3153$ observation matrix, including logistic regression, support vector machines, K-Nearest Neighbor (KNN), ensemble methods, etc. After much experimentation and fine tuning, based on 10-fold cross validation runs and on random tests carried out by the Challenge organizers on hidden data, the best performance was obtained with a KNN classifier with the following parameters:

- Core tensor dimensions: $23 \times 4 \times 5$.
- $F = 45$ and K $= 10$.
- Inverse-squared of Spearman distance.
- Feeding of features and its squares (pure quadratic).
- Standardized features (zero mean, unit variance).
- Weighted observations (cubic root of the number of complete heart beats).

The problem of imbalanced data was dealt with by shifting the decision boundary on the posterior probability (scores) given by the classifier, chosen to be equal to the minority class proportion, that is $665 \div 3153 = 0.211$. Therefore, observations with posterior probability greater than 0.211 were classified as Abnormal. Table 2 summarizes the classifier performance in a random, 10-fold cross validation (average of 3 runs), and on hidden test data (1,277 recordings). This resulted in the $\mathbf{4^{th}}$ top performance among the Challenge participants.

In order to evaluate the generalization error, this approach was tried on different aggregations of databases. First, in Table 3, we compare a 10-fold CrossVal with only database *e* as training set, vs one with all databases but *e*. Then, in Table 4, a classifier was trained by leaving out one database at a time, on which it was then tested.

One can conclude that the approach generalizes poorly when a database is not represented in the training set, and that the contribution of database *e* is key in achieving a good performance on the whole set. This also suggests

Test DB	Bal. Acc.	Se	Sp
a	56.2%	79.1%	33.3%
b	51.8%	37.5%	71.2%
c	54.1%	95.7%	12.5%
d	59.3%	96.4%	22.2%
e	48.0%	77.6%	18.3%
f	50.6%	100.0%	1.3%

Table 4. Leave-out test on different databases.

that in order to obtain reliable results with a practical devices, both the instrumentation/sensor and the recording procedure will demand some degree of standardization.

Note: Author is a Visiting Scholar at MIT, affiliated to IAE Business School, Austral University, Argentina, email: idiazbobillo@iae.edu.ar.

References

[1] Alam U, Asghar O, Khan SQ, Hayat S, Malik RA. Cardiac auscultation: an essential clinical skill in decline. Br J Cardiol 2010;17(1):8–10.

[2] Leng S, San Tan R, Chai KTC, Wang C, Ghista D, Zhong L. The electronic stethoscope. Biomedical engineering online 2015;14(1):1.

[3] Liu C, Springer D, Li Q, Moody B, Juan RA, Chorro FJ, Castells F, Roig JM, Silva I, Johnson AE, Syed Z, Schmidt SE, Papadaniil CD, Hadjileontiadis L, Naseri H, Moukadem A, Dieterlen A, Brandt C, Tang H, Samieinasab M, Samieinasab MR, Sameni R, Mark RG, Clifford GD. An open access database for the evaluation of heart sound algorithms. Physiological Measurement 2016;37(9).

[4] Goovaerts G, De Wel O, Vandenberk B, Willems R, Van Huffel S. Detection of irregular heartbeats using tensors. In 2015 Computing in Cardiology Conference (CinC). IEEE, 2015; 573–576.

[5] Phan AH, Cichocki A. Tensor decompositions for feature extraction and classification of high dimensional datasets. Nonlinear theory and its applications IEICE 2010;1(1):37–68.

[6] Kolda TG, Bader BW. Tensor decompositions and applications. SIAM review 2009;51(3):455–500.

[7] De Lathauwer L, Vandewalle J. Dimensionality reduction in higher-order signal processing and rank-(r 1, r 2,, r n) reduction in multilinear algebra. Linear Algebra and its Applications 2004;391:31–55.

[8] Springer DB, Tarassenko L, Clifford GD. Logistic regression-hsmm-based heart sound segmentation. IEEE Transactions on Biomedical Engineering 2016;63(4):822–832.

[9] Brookes M. Voicebox: Speech processing toolbox for matlab, 2016. URL `http://www.ee.ic.ac.uk/...hp/staff/dmb/voicebox/voicebox.html`. Available online.

[10] Vervliet N, Debals O, Sorber L, Van Barel M, De Lathauwer L. Tensorlab 3.0, 2016. URL `http://www.tensorlab.net`. Available online.