SPEECH DENOISING BASED ON A GREEDY ADAPTIVE DICTIONARY ALGORITHM

Maria G. Jafari and Mark D. Plumbley

Queen Mary University of London.
Centre for Digital Music
Mile End Road, London E1 4NS, UK
Email: maria.jafari@elec.qmul.ac.uk; mark.plumbley@elec.qmul.ac.uk
Web: http://www.elec.qmul.ac.uk/digitalmusic/

ABSTRACT

In this paper we consider the problem of speech denoising based on a greedy adaptive dictionary (GAD) algorithm. The transform is orthogonal by construction, and is found to give a sparse representation of the data being analysed, and to be robust to additive Gaussian noise.
The performance of the algorithm is compared to that of the principal component analysis (PCA) method, for a speech denoising application. It is found that the GAD algorithm offers a sparser solution than PCA, while having a similar performance in the presence of noise.

1. INTRODUCTION

Speech signals are often degraded by the presence of noise, arising for instance from the recording equipment or the surrounding environment. In some applications it may be desirable to reduce the noise level by applying some form of pre-processing. As a result, several denoising algorithms exist, including general methods such as wavelet denoising [1], and principal component analysis [2], and others dedicated to audio signals, such as methods based on time-frequency representations [3] or sparse linear regression [4].
Sparse methods, in particular, are well suited to the analysis of speech and music signals, and have acquired great popularity in recent years. Sparse representations, in which most coefficients are close to zero, are used extensively because they allow the information within a signal to be conveyed with only a few elementary components, denoted as atoms, which are obtained from the decomposition. Moreover, they often help uncover hidden structure in the analysed signal. Sparsification of a signal is often carried out using overcomplete dictionaries, in which the number of atoms is greater than the dimensionality of the signal space [5]. It is generally accepted that overcomplete dictionaries allow the achievement of higher sparsity than could be obtained with orthogonal transforms, thanks to the range of waveforms present in the dictionary to match the signal features [6]. The aim is then to find, among the many possible representations for the signal of interest, one with a small number of significant coefficients. This is a non-trivial problem that has been shown to be NP-hard [7].

Orthonormal linear transforms such as the wavelet and short-time Fourier transform have also been used to sparsify the signal, and have the advantage of being easily invertible, since if the matrix $\mathbf{T}$ is orthonormal, then $\mathbf{T}\mathbf{T}^T = \mathbf{I}$. Thus, the number of atoms within the dictionary equals the dimension of the signal space, leading to a unique signal representation.

In [8], we addressed the problem of representing a speech signal using an orthogonal sparsifying transform. The adaptive transform is based on a greedy algorithm, which learns a dictionary from the observed data, re-arranged into frames. The algorithm maximizes the L2-norm of the data, while minimizing its L1-norm. The transform is forced to be orthogonal by removing all the components lying in the direction of a particular vector, corresponding to the selected data frame, at each iteration.
There are two main advantages to this orthonormal greedy adaptive dictionary (GAD) algorithm: firstly, the atoms are extracted from the observed data, and therefore they are directly relevant to the data being analysed; secondly, the fact that the transform is orthogonal, implies that only direct matrix multiplication will be needed to analyze the signal. The algorithm in [8] was subsequently extended in [9] to address the problem of source separation in echoic and anechoic environments.

In this paper, we consider the application of the GAD algorithm to the problem of speech denoising. We compare the results for the GAD transform with those obtained with dictionaries learned using principal component analysis, which is chosen because the approach followed by the two algorithms is somewhat similar. PCA is a statistical technique that finds from the data the principal components that maximise the variances. Then, the components corresponding to the highest variances are retained to represent the signal, while those corresponding to low variances relate to the noise, and can be omitted in order to reduce the noise level. Similarly, the GAD method extracts components one at the time, by selecting the atoms with highest L2-norm first, and those with lower L2-norm are found to correspond to noise. Experimental results show that the GAD algorithm performs denoising in a manner similar to PCA, generally resulting in better noise reduction than the latter.

The structure of the paper is as follows: the problem that we seek to address is outlined in section 2, the greedy adaptive dictionary algorithm is summarised in section 3, and the principal component analysis method is outlined in section 4. Experimental results for the speech representation and denoising applications are presented in section 5. Finally, conclusions are drawn in section 6.

2. PROBLEM STATEMENT

The problem addressed here is that of learning a dictionary $\mathcal{D}$ consisting of L atoms $\boldsymbol{\psi}^l(n)$, that is $\mathcal{D} = \{\boldsymbol{\psi}^l(n)\}_{l=1}^L$,

which provides a sparse representation of a real-valued noisy observed signal $\mathbf{x}(n) = [x(1), \ldots, x(N)]^T$

$$\mathbf{x}(n) = \mathbf{s}(n) + \mathbf{v}(n) \tag{1}$$

where $L \ll N$, and $\mathbf{v}(n)$ represents the noise vector.

The GAD method does not operate upon all the signal samples, but the observed data is divided into blocks, obtained when overlapping data frames, with an overlap of T samples, are taken from $\mathbf{x}(n)$. The new signal blocks are denoted as $\mathbf{x}_k(n) = [x_k(1), \ldots, x_k(L)]^T$. The result is a newly constructed matrix $\mathbf{X}(n) = [\mathbf{x}_1(n), \mathbf{x}_2(n), \ldots, \mathbf{x}_K(n)]$, whose k-th column is represented by the signal block $\mathbf{x}_k(n)$.

The dictionary is learned from the signals in the columns of $\mathbf{X}(n)$, so that the problem can now be stated as follows: given a real valued signal $\mathbf{x}_k(n) = [x_k(1), \ldots, x_k(L)]^T$, and an orthogonal dictionary $\mathcal{D} = \{\boldsymbol{\psi}^l(\tau)\}_{l=1}^L$, we seek a decomposition of $\mathbf{x}_k(n)$, such that [10]

$$\mathbf{x}_k(n) = \sum_{l=1}^{L} \alpha_k^l \boldsymbol{\psi}^l(n), \ \forall k \in \{1, \ldots, K\} \tag{2}$$

where α^l are the expansion coefficients which encode explicit information regarding the properties of the signal $\mathbf{x}_k(n)$, depending on the choice of dictionary $\mathcal{D}$.

3. GREEDY ADAPTIVE DICTIONARY ALGORITHM (GAD)

The GAD algorithm adaptively learns a data dependent dictionary by sequentially extracting the columns of the matrix $\mathbf{X}$, where the time index has been dropped for the sake of clarity. At each iteration, extraction of a new atom depends on finding the column of $\mathbf{X}$ that satisfies:

$$\max_k \frac{||\mathbf{x}_k||_2}{||\mathbf{x}_k||_1} \tag{3}$$

where $||\cdot||_1$ and $||\cdot||_2$ denote the L1- and L2-norm respectively. Thus at each iteration, the method reduces the energy of the data by a maximum amount, across all frames, while ensuring that the L1-norm is reduced by a minimum amount.

The algorithm solves the maximization problem in equation (3) according to the following steps:

1. Initialisation:
   - ensure that the columns of $\mathbf{X}$ have unit L1-norm

     $$\tilde{\mathbf{x}}_k = \frac{\mathbf{x}_k}{\|\mathbf{x}_k\|_1} \tag{4}$$

     where $\mathbf{x}_k$ the k-th column of $\mathbf{X}$. This leads to a new data matrix $\tilde{\mathbf{X}}$, whose columns now have unit L1-norm.
   - Initialise the residual matrix

     $$\mathbf{R}^0 = \tilde{\mathbf{X}} \tag{5}$$

     where $\mathbf{R}^j = [\mathbf{r}_1^j, \ldots, \mathbf{r}_K^j]$, and $\mathbf{r}_k^j$ is a K-dimensional residual column vector corresponding to the k-th column of $\mathbf{R}^j$.
2. Compute the L2-norm of each frame

   $$E_k = ||\mathbf{r}_k^j||_2 = \sum_{n=1}^{L} |\mathbf{r}_k^j(n)|^2. \tag{6}$$

3. Find the index $\hat{k}$ corresponding to the signal block with largest L2-norm, $\mathbf{r}_{\hat{k}}^j$

   $$\hat{k} = \arg\max_{k \in \mathbb{K}} (E_k) \tag{7}$$

   where $\mathbb{K} = \{1, \ldots, K\}$ is the set of all indices pointing to the columns of $\mathbf{R}^j$.

At each iteration $j \in \{1, \ldots, L\}$, the signal with highest L2-norm, $\mathbf{r}_{\hat{k}}^j$, becomes a dictionary element, and we iteratively define a residual matrix $\mathbf{R}^j \in \mathbb{R}^{L \times K}$, which decreases by the appropriate amount, determined by the selected atom $\boldsymbol{\psi}^j$ and the coefficient of expansion α_k^j.

4. Set the j-th dictionary element $\boldsymbol{\psi}^j$ to be equal to the signal block with largest L2-norm $\mathbf{r}_{\hat{k}}^j$

   $$\boldsymbol{\psi}_j = \frac{\mathbf{r}_{\hat{k}}^j}{||\mathbf{r}_{\hat{k}}^j||_1}. \tag{8}$$

5. Evaluate the coefficients of expansion, given by the inner product between the residual vector $\mathbf{r}_k^j$, and the atom $\boldsymbol{\psi}^j$

   $$\alpha_k^j = \langle \mathbf{r}_k^j, \boldsymbol{\psi}^j \rangle. \tag{9}$$

6. Compute the new residual, by removing the component along the chosen atom, for each element k in $\mathbf{r}_k^j(n)$

   $$\mathbf{r}_k^{j+1} = \mathbf{r}_k^j - \frac{\alpha_k^j}{\langle \boldsymbol{\psi}^j, \boldsymbol{\psi}^j \rangle} \boldsymbol{\psi}^j. \tag{10}$$

The term in the denominator of $\frac{\alpha_k^j}{\langle \boldsymbol{\psi}^j, \boldsymbol{\psi}^j \rangle}$ in equation (10), is included to ensure that the coefficient of expansion α_k^j corresponding to the inner product between the selected atom $\boldsymbol{\psi}^j$ and the frame of maximum L2-norm $\mathbf{r}_{\hat{k}}^j$, is normalised to 1. Then, the corresponding column of the residual matrix $\mathbf{R}_j$ is set to zero, since the whole atom is removed. This ensures that the transform is orthogonal. Finally, the signal matrix is updated by the residual, and the whole process is repeated:

7. Repeat from step 2.

A clear advantage of the GAD algorithm is that it results in an orthogonal transform, and therefore the inverse transform of $\mathbf{Y}$is evaluated straightforwardly from $\mathbf{X}^L = \mathbf{DY}$, where $\mathbf{X}^L$ is the L term approximation of the signal $\mathbf{X}$, and $\mathbf{D} = [(\boldsymbol{\psi}^1)^T, \ldots, (\boldsymbol{\psi}^{k\max})^T]$ is the dictionary matrix.

4. PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal component analysis seeks to remove the correlation from the observed signals by finding the projections of the data in the directions of maximum variances [11]. It corresponds to the eigenvalue decomposition of the correlation

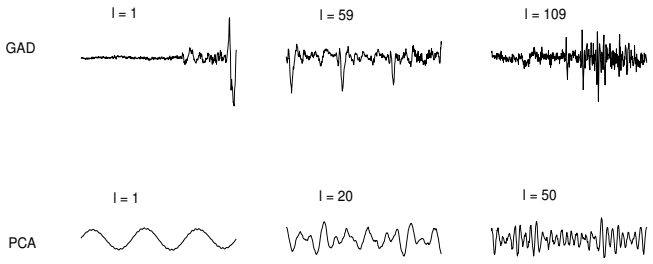

Figure 1: Examples of the atoms learned with the GAD and PCA algorithms. The letter 'l' in the plots denotes the position of the atom within the dictionary.

matrix $\mathbf{R}_{xx}$=$E\left\{\mathbf{x}_k\mathbf{x}_k^T\right\}$ of the L-dimensional data vector $\mathbf{x}_k$:

$$\mathbf{R}_{xx}= \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1} \tag{11}$$

where $\mathbf{\Lambda}$= diag $\{\lambda_1, \ldots, \lambda_L\}$ is a diagonal matrix whose entries are the eigenvalues of $\mathbf{R}_{xx}$, arranged in decreasing order: $\lambda_1 > \lambda_2 > \ldots > \lambda_L$, such that $\lambda_1 = \lambda_L$, and $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_L]$ is the matrix whose columns are the associated eigenvectors [2]. PCA seeks a linear transform $\mathbf{W}_{PCA} = \mathbf{\Lambda}^{1/2}\mathbf{Q}^T$, such that the transformed signals are uncorrelated, the basis vectors are orthogonal to each other, and the eigenvalues are ordered. The resulting eigenvectors represent the principal directions along which the variances are maximised, and the eigenvalues define the values of the variances. The basis vectors for PCA, therefore, vary from signal to signal. The vector $\mathbf{z}_k$, representing the projections of the column vector $\mathbf{x}_k$ onto the principal directions, is given by

$$\mathbf{z}_k = \mathbf{W}_{PCA}\mathbf{x}_k \tag{12}$$

The elements of the output vector $\mathbf{z}_k$ are now uncorrelated, and are referred to as the principal components. The main purpose of PCA is to reduce the dimension of the data, that is, only the components that have large variances are retained [2]. This decreases the computational cost of subsequent processing steps, and leads to noise reduction, because PCA decomposes mixed signals into two subspaces: the signal subspace spanned by those components associated with the largest eigenvalues, and the noise subspace formed by the components corresponding to the smallest eigenvalues [12].

5. APPLICATIONS

5.1 Speech Representation

In this section we compare the GAD and PCA algorithms, for the analysis of a male speech signal, and in both cases we look for a dictionary containing 512 atoms.

Figure 1 shows examples of the atoms learned with GAD and PCA, respectively. It was observed that the atoms extracted with PCA are not particularly localized, and do not appear to be capturing any specific features of the speech signal, but perhaps more general characteristics. The GAD algorithm yields atoms that appear to capture more information about the signal, and which are fairly localized.

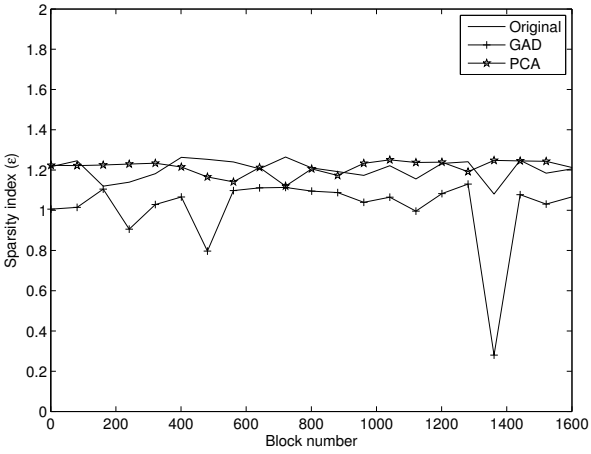

Figure 2: Sparsity index for the GAD algorithm, compared to the original signal and PCA.

Method	Number of Atoms					
	512	400	300	200	100	50
GAD	0.0	0.7	1.7	2.8	5.3	6.8
PCA	0.0	0.2	0.6	1.3	2.5	4.0

Table 1: Approximation error $\epsilon \times 10^{-3}$ for the GAD and PCA algorithms. All values are expressed in decibels (dB).

To determine how sparse the representation obtained with the proposed approach is, we plot the sparsity index for the transform coefficients obtained with the two methods. The sparsity index of a signal y as

$$\xi = ||y||_1/||y||_2 \tag{13}$$

generally, the lower the sparsity index is, the sparser the signal y. Figure 2 shows a plot of the sparsity index for the original signal blocks in $\mathbf{X}$, and for the coefficients of expansion obtained with the GAD and PCA algorithms. We can see that the signal transformed with the GAD algorithm is sparser than in the time domain, and than the coefficients obtained with PCA. Next, we consider the accuracy of the approximation by looking at the approximation error ϵ obtained when the function f is approximated by $\tilde{f}$,

$$\epsilon = ||\tilde{f} - f||_2 \tag{14}$$

Table 1 shows the error, $\epsilon \times 10^{-3}$, for both algorithms describing the accuracy of the approximation as the number of atoms used in the signal reconstruction decreases from 512 to 50. The results indicate that PCA performs best, because the transform arranges the signal components so that most energy is concentrated in a small number of components, typically corresponding to those extracted earlier. The approach followed by the GAD transform is somewhat similar, since the atoms with highest L2-norm are extracted first, and therefore the algorithm also results in good signal approximations as the number of atoms is reduced.

SNR	Method	Number of Atoms					
		512	400	300	200	100	50
20 dB	GAD	0.0	1.0	1.7	1.3	-1.9	-6.4
	PCA	0.0	0.6	1.3	1.8	-0.4	-4.9
10 dB	GAD	0.0	1.3	2.5	3.9	4.5	2.4
	PCA	0.0	0.5	1.3	2.8	4.9	3.8
5 dB	GAD	0.0	1.3	2.4	4.1	5.8	5.6
	PCA	0.0	0.5	1.3	2.6	5.1	6.6
0 dB	GAD	0.0	1.3	2.6	4.4	7.0	8.1
	PCA	0.0	0.5	1.3	2.7	5.4	7.9
-5 dB	GAD	0.0	1.4	2.8	4.9	7.9	9.9
	PCA	0.0	0.5	1.3	2.7	5.4	8.3
-10 dB	GAD	0.0	1.3	2.9	4.9	7.9	10.5
	PCA	0.0	0.5	1.3	2.7	5.3	8.0

Table 2: ISNR for the GAD and PCA algorithms. All values are expressed in decibels (dB).

5.2 Speech Denoising

To evaluate the effect of the algorithms over the observed noisy data, we consider the ISNR:

$$ISNR = 10log\frac{E\{(s-x)^2\}}{E\{(s-\hat{s})^2\}} \quad (15)$$

where s is the original signal, x is the observed distorted (noisy) signal, and $\hat{s}$ is the source approximated by the transform. As the signal approximation becomes closer to the original source, ISNR increases.

The ISNR for the GAD and PCA methods is shown in Table 2, for a noise level changing from 20 dB to -10 dB, as the number of atoms in the reconstruction is reduced from 512 to 50. The noise levels and number of atoms used in the reconstruction were selected via an informal listening test, which indicated that a lower signal to noise ratio or fewer atoms resulted in very poor signal quality. The underlined values correspond to the highest ISNR achieved by the algorithm for a particular noise level.

When all atoms are used in the reconstruction, both the GAD and PCA transforms yield an ISNR of 0 dB. When the noise is low (20 dB), reducing the number of atoms in the reconstruction leads to distortion in the signal approximation, yielding negative results in ISNR. As the level of noise increases, the high ISNR values for PCA and GAD indicate that there are benefits in reducing the number of atoms used in the signal approximation. It is well-known that PCA can reduce the level of noise present, because it decomposes the space into signal and noise subspaces. The results in table 2 show that, similarly, GAD achieve a reduction in the noise level. It is also evident that the GAD method generally yields a higher improvement in SNR than PCA, especially as the noise level increases, and fewer atoms are used in the approximation.

6. CONCLUSIONS

In this paper we have presented a speech denoising method based on a greedy orthogonal adaptive dictionary learning algorithm, which we have shown to result in sparse representations for speech signals. The algorithm constructs a user-defined complete dictionary, whose atoms clearly encode local properties of the signal.

The performance of the algorithm was compared to that of the PCA method, and it was found to give good signal approximations, even as the number of atoms in the reconstructions decreases considerably; it was also observed that the algorithm has good tolerance to noise, comparable to that afforded by PCA.

In future work, we will be benchmarking the performance of the GAD algorithm against other denoising algorithms. We will also be considering artefacts and signal distortion.

REFERENCES

[1] M. Vetterli, "Wavelets, approximation, and compression," *IEEE Signal Processing Magazine*, vol. 18, pp. 59–73, 2001.

[2] S. Haykin, *Neural networks: a comprehensive foundation*, Prentice Hall, 2nd edition, 1999.

[3] G. Yu, S. Mallat, and E. Bacry, "Audio denoising by time-frequency block thresholding," *IEEE Trans. on Signal Processing*, vol. 56, pp. 1830–1839, 2008.

[4] C. Fevotte, B. Torresani, L. Daudet, and S. Godsill, "Sparse linear regression with structured priors and application to denoising of musical audio," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, pp. 174–185, 2008.

[5] P. Huggins and S. Zucker, "Greedy basis pursuit," *IEEE Transactions on Signal Processing*, vol. 55, pp. 3760–3772, 2007.

[6] R. Gribonval and S. Lesage, "A survey of sparse component analysis for blind source separation: principles, perspectives, and new challanges," in *Proceedings of the 2006 European Symposium on Artificial Neural Networks (ESANN '06)*, 2006, pp. 323–330.

[7] G. Davis, *Adaptive nonlinear approximations*, Ph.D. thesis, New York University, 1994.

[8] M. Jafari and M. Plumbley, "An adaptive orthogonal sparsifying transform for speech signals," in *Proc. of the Int. Symposium on Communications, Control and Signal Processing (ISCCSP '08)*, 2008, pp. 786–790.

[9] M. Jafari and M. Plumbley, "Separation of stereo speech signals based on a sparse dictionary algorithm," in *Proc. of the European Signal Processing Conference (EUSIPCO '08)*, 2008.

[10] M. Goodwin and M. Vetterli, "Matching pursuit and atomic signal models based on recursive filter banks," *IEEE Trans. on Signal Processing*, vol. 47, pp. 1890–1902, 1999.

[11] A. Hyvärinen, "Survey on independent component analysis," *Neural Computing Surveys*, vol. 2, pp. 94–128, 1999.

[12] S. Haykin, Ed., *Unsupervised Adaptive Filtering. Volume I: Blind Source Separation*, John Wiley & Sons, 2000.