A Polynomial Dictionary Learning Method for Acoustic Impulse Response Modeling

Jian Guan[1], Jing Dong[2], Xuan Wang[1], and Wenwu Wang[2]

[1] Computer Application Research Center
Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, 518055, China
[2] University of Surrey, Guildford, GU2 7XH, United Kingdom
Emails: [j.guan; wangxuan]@cs.hitsz.edu.cn; [j.dong; w.wang]@surrey.ac.uk

Abstract. Dictionary design is an important issue in sparse representations. As compared with pre-defined dictionaries, dictionaries learned from training signals may provide a better fit to the signals of interest. Existing dictionary learning algorithms have focussed overwhelmingly on standard matrix (i.e. with scalar elements), and little attention has been paid to polynomial matrix, despite its widespread use for describing convolutive signals and for modelling acoustic channels in both room and underwater acoustics. In this paper, we present a method for polynomial matrix based dictionary learning by extending the widely used K-SVD algorithm to the polynomial matrix case. The atoms in the learned dictionary form the basic building components for the impulse responses. Through the control of the sparsity in the coding stage, the proposed method can be used for denoising of acoustic impulse responses, as demonstrated by simulations for both noiseless and noisy data.

Keywords: Dictionary learning, polynomial matrix, impulse responses

1 Introduction

Sparse representation has drawn intensive research interest for more than a decade. It aims to represent a signal with a linear combination of a small number of atoms chosen from an overcomplete dictionary in which the total number of atoms is greater than the dimension of the signal atoms [5]. The dictionary can be pre-defined using a mathematical equation. Alternatively, it can also be adapted from training data with a machine learning algorithm, leading to a category of algorithms called dictionary learning.

Several algorithms have been proposed for dictionary learning, such as the MOD, K-SVD and SimCO algorithms [1], [3]. These algorithms have shown promising results in a number of tasks (e.g. denoising, super-resolution, and source separation) for a variety of natural signals, including acoustic and image data. In some practical applications, however, these algorithms cannot be directly applied since the signals that need to be dealt with may contain time delays. For example, in acoustic modelling, the propagation channels between sources and microphones (or hydrophones) are often represented as acoustic impulse responses, which are usually described by polynomial matrix (with time

delays) rather than a standard matrix with scalar elements. Polynomial matrices have been widely used in acoustic and communication channel modelling [7], e.g. for convolutive mixing and unmixing. Each element in a polynomial matrix can be represented as a finite impulse response (FIR) filter, e.g. for describing the input-output relationship [6].

In this paper, we develop a polynomial dictionary learning technique based on K-SVD algorithm. More specifically, we extend the K-SVD algorithm [1] for polynomial matrix based sparse representation model. Each atom in the learned dictionary is a polynomial represented as a FIR filter. All the atoms in the dictionary provides an overall description of the acoustic environment from which the acoustic impulse responses are used to train the dictionary. Such a dictionary has potential applications in denoising, dereverberation/deconvolution, and channel shortening of acoustic impulse responses, as partly demonstrated by simulations.

The remainder of the paper is organized as follows: Section 2 gives a brief review of the background of conventional dictionary learning and polynomial matrix decomposition. Section 3 presents the proposed polynomial dictionary learning method in detail. Section 4 shows the simulations and results. Section 5 concludes this paper.

2 Background

2.1 Conventional Dictionary Learning

Given a signal $\mathbf{y} \in \mathbb{R}^n$, the sparse representation of $\mathbf{y}$ can be expressed as

$$\mathbf{y} = \mathbf{D}\mathbf{x} \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{n \times K}$ ($n \ll K$) is an overcomplete dictionary containing K atoms, $\{\mathbf{d}_j\}_{j=1}^{K} \in \mathbb{R}^n$, and $\mathbf{x} \in \mathbb{R}^K$ is the sparse coefficient vector for representing $\mathbf{y}$. Two problems are often studied, namely, sparse coding and dictionary learning. Sparse coding aims to estimate $\mathbf{x}$, given $\mathbf{y}$ and $\mathbf{D}$, subject to the constraint that $\mathbf{x}$ is sparse, i.e. the number of non-zero elements measured by l_0 norm is below a pre-defined threshold (or relaxed by the l_1 norm of $\mathbf{x}$). In dictionary learning, the aim is to train a dictionary $\mathbf{D}$ based on a set of signals $\{\mathbf{y}_i\}_{i=1}^{N}$ which form a matrix $\mathbf{Y} \in \mathbb{R}^{n \times N}$, subject to the constraint that sparse coding coefficient matrix $\mathbf{X}$ is sparse. Here we focus on the dictionary learning problem.

2.2 Polynomial Matrix

A polynomial matrix is a matrix whose elements are polynomials. A $p \times q$ polynomial matrix $\mathbf{A}(z)$ can be expressed as

$$\mathbf{A}(z) = \sum_{\ell=0}^{L-1} \mathbf{A}(\ell) z^{-\ell} = \begin{bmatrix} a_{11}(z) & a_{12}(z) & \cdots & a_{1q}(z) \\ a_{21}(z) & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{p1}(z) & \cdots & \cdots & a_{pq}(z) \end{bmatrix} \tag{2}$$

where $\mathbf{A}(\ell) \in \mathbb{C}^{p\times q}$ is the coefficient matrix of $z^{-\ell}$, which denotes the impulse response at lag ℓ, and L is the length of impulse response. We use the coefficient of the polynomial to express the magnitude of the impulse responses. The F-norm of the polynomial matrix is defined as

$$\|\mathbf{A}(z)\|_F = \sqrt{\sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{\ell=0}^{L-1} |a_{ij}(\ell)|^2} \tag{3}$$

Polynomial matrices have been widely used for acoustic impulse response modelling to describe the multi-path channel impulses propagating from the sources to the sensors. There are other forms of polynomial matrix decomposition techniques such as polynomial eigen-value or singular-value decompositions [4]. In this paper, we develop a polynomial dictionary learning algorithm by extending the conventional dictionary learning algorithm to the polynomial matrix case, as detailed next.

3 Polynomial Dictionary Learning

The conventional dictionary learning model (1) can be extended to the polynomial case as

$$\mathbf{Y}(z) = \mathbf{D}(z)\mathbf{X} \tag{4}$$

where $\mathbf{D}(z) \in \mathbb{C}^{n\times K}$ is an overcomplete polynomial dictionary matrix which contains polynomial atoms, $\mathbf{Y}(z) \in \mathbb{C}^{n\times N}$ is the "signals" to be represented, which can be an impulse response matrix, and $\mathbf{X} \in \mathbb{R}^{K\times N}$ is the sparse matrix which contains the representation coefficients of $\mathbf{Y}(z)$.

Suppose the length of the impulse response in $\mathbf{Y}(z)$ is L, according to equation (2), model (4) can be represented as

$$\sum_{\ell=0}^{L-1} \mathbf{Y}(\ell)z^{-\ell} = \sum_{\ell=0}^{L-1} \mathbf{D}(\ell)z^{-\ell}\mathbf{X} \tag{5}$$

where $\mathbf{Y}(\ell) \in \mathbb{R}^{n\times N}$ and $\mathbf{D}(\ell) \in \mathbb{R}^{n\times K}$ are the coefficient matrices of polynomial matrix $\mathbf{Y}(z)$ and $\mathbf{D}(z)$ at lag ℓ, respectively. As can be seen from equation (5), for any $\ell \in (0, L-1)$, $\mathbf{Y}(\ell)$ is represented as the linear combination of atoms in $\mathbf{D}(\ell)$, and $\mathbf{X}$ is the representation coefficients. This means that each coefficient matrix of the polynomial matrix $\mathbf{Y}(z)$ at different lags can also be represented by a coefficient matrix of $\mathbf{D}(z)$ at the corresponding lag weighted by the same representation matrix $\mathbf{X}$. Therefore, the coefficient matrices of polynomial matrices $\mathbf{Y}(z)$ and $\mathbf{D}(z)$ satisfy the form

$$\mathbf{Y}_i = \mathbf{D}_i\mathbf{X} \tag{6}$$

where $\mathbf{Y}_i \in \mathbb{R}^{n\times N}$ and $\mathbf{D}_i \in \mathbb{R}^{n\times K}$ are the coefficient matrices of $\mathbf{Y}(z)$ and $\mathbf{D}(z)$ at lag $i = 0, 1, ..., L-1$, respectively. Note that, for notational convenience, we

have used $\mathbf{Y}_i$ to denote the coefficient matrix $\mathbf{Y}(\ell)$. We then define the following matrices by concatenating the coefficient matrices at all the time lags vertically

$$\underline{\mathbf{Y}} = [\mathbf{Y}_0; \mathbf{Y}_1; \mathbf{Y}_2; ...; \mathbf{Y}_{L-1}] \tag{7}$$

$$\underline{\mathbf{D}} = [\mathbf{D}_0; \mathbf{D}_1; \mathbf{D}_2; ...; \mathbf{D}_{L-1}] \tag{8}$$

As a result, equation (4) can be rewritten as

$$\underline{\mathbf{Y}} = \underline{\mathbf{D}}\mathbf{X} \tag{9}$$

where $\underline{\mathbf{Y}} \in \mathbb{R}^{nL \times N}$ and $\underline{\mathbf{D}} \in \mathbb{R}^{nL \times K}$. We can see from equation (9) that the polynomial dictionary learning problem (4) is now converted to a conventional dictionary learning model. Similarly, $\underline{\mathbf{D}}$ is overcomplete, hence, nL should be much smaller than K in equation (9).

The new dictionary $\underline{\mathbf{D}}$ can now be learned with a conventional dictionary learning algorithm such as the K-SVD algorithm [1], in which each atom and its corresponding sparse coefficient are updated simultaneously one by one with an iterative process. The learned $\underline{\mathbf{D}}$ can be used to reconstruct $\underline{\mathbf{Y}}$, and the reconstructed matrix is denoted as $\underline{\hat{\mathbf{Y}}}$. With a reverse operation to equation (7), we can obtain the coefficient matrix of the polynomial matrix at each time lag, as follows

$$\underline{\hat{\mathbf{Y}}} = \left[\hat{\mathbf{Y}}_0; \hat{\mathbf{Y}}_1; \hat{\mathbf{Y}}_2; ...; \hat{\mathbf{Y}}_{L-1}\right] \tag{10}$$

where $\underline{\hat{\mathbf{Y}}}$ is the restored matrix of $\underline{\mathbf{Y}}$, $\hat{\mathbf{Y}}_i$ is the coefficient matrix of polynomial channel matrix $\underline{\hat{\mathbf{Y}}}(z)$ at lag i, $i = 0, 1, 2..., L-1$. With the coefficient matrices obtained above we can then construct the polynomial matrix $\underline{\hat{\mathbf{Y}}}(z)$ by using equation (2).

4 Simulations and Results

In this section, we evaluate the performance of the proposed method for learning a polynomial dictionary, and use it to recover a polynomial matrix. Two types of polynomial matrices were used, with one (i.e. the elements of its coefficient matrix) generated randomly, and the other as acoustic impulse responses (using a room image model). In both cases, noise is added to evaluate the capability of the proposed method for the recovery of noisy acoustic impulse responses.

4.1 Data Generation and Performance Measure

Polynomial Matrices Synthesis The polynomial matrices were generated synthetically as follows. First, we generated a random scalar matrix $\underline{\mathbf{D}}$ with uniformly distributed entries, which was then used as the coefficient matrix for the polynomial matrix $\mathbf{D}(z)$ where each column of $\underline{\mathbf{D}}$ was normalized. Then, $\underline{\mathbf{Y}}$ was generated by the linear combination of different columns in $\underline{\mathbf{D}}$. At last, the polynomial matrices $\mathbf{Y}(z)$ and $\mathbf{D}(z)$ were generated by splitting their coefficient matrices according to equation (7) and (8). The dimensions of the signals and dictionaries are designed according to the different experiments described later in this section.

Acoustic Impulse Response Generation and Modeling The acoustic impulse responses were generated in a $20 \times 20 \times 3$ m^3 room (to simulate a large hall) by the image model [2], where the reverberation time is 900 ms, sampling frequency is 16 KHz. The number of sampling points is set to be 14400 which means the number of time lags for each impulse response is 14400, and 1600 acoustic impulse responses were generated as the training set. We use a polynomial matrix to model acoustic signals by splitting each acoustic impulse response signal into 80 sections, where each section was modeled as a polynomial with 20 lags, so all these 1600 room acoustic impulse responses can be modeled as a 10×115200 polynomial matrix with 20 time lags for each element.

Performance Index We define two performance indices to measure how well our proposed method performs. The *error rate* of the recovered polynomial matrix is defined as

$$E_r = \frac{\|\mathbf{Y}(z) - \hat{\mathbf{Y}}(z)\|_F^2}{\|\mathbf{Y}(z)\|_F^2} \tag{11}$$

where $\mathbf{Y}(z)$ denotes the original polynomial matrix without noise, and $\hat{\mathbf{Y}}(z)$ is the recovered polynomial matrix of $\mathbf{Y}(z)$. E_r means how similar the recovered $\hat{\mathbf{Y}}(z)$ to $\mathbf{Y}(z)$, the smaller the better. We also defined an *error rate* of noisy signal

$$E_n = \frac{\|\mathbf{Y}(z) - \mathbf{Y}_n(z)\|_F^2}{\|\mathbf{Y}(z)\|_F^2} \tag{12}$$

where $\mathbf{Y}_n(z)$ is the noisy polynomial matrix obtained by adding noise to $\mathbf{Y}(z)$, and E_n reflects the difference between $\mathbf{Y}(z)$ and $\mathbf{Y}_n(z)$ (i.e. the relative noise level).

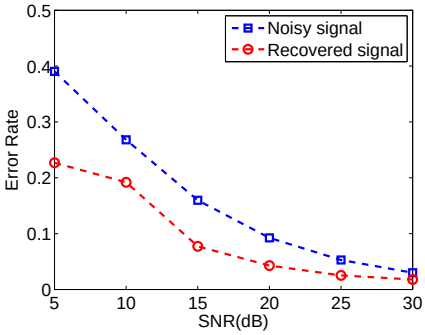

Fig. 1: The error rate comparison between E_n and E_r for the test signals at different SNR levels.

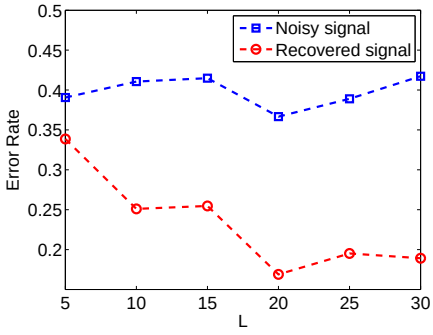

Fig. 2: The error rate comparison of E_n and E_r at different time lags for the test signal with the same noise level.

4.2 Experiments

We carry out several experiments on both synthetic data and simulated room acoustic impulse responses. The proposed method is tested on different noise levels, different impulse response lags, and the recovery of acoustic impulse response from noisy data.

Experiments on Synthetic Impulse Response Data We synthesized a 10×1500 polynomial matrix $\mathbf{Y}(z)$ with 3 lags as training data. The dictionary $\mathbf{D}(z)$ was 10×50 polynomial matrix. The sparsity was set to be 3. In order to test whether our method can recover a signal (i.e. polynomial matrix) corrupted by noise of different levels, white Gaussian noise at different signal-to-noise (SNR) ratios was added to the test signal. Note that noise was added to the coefficient matrix of the polynomial matrix. Figure 1 shows how the error rate changes at

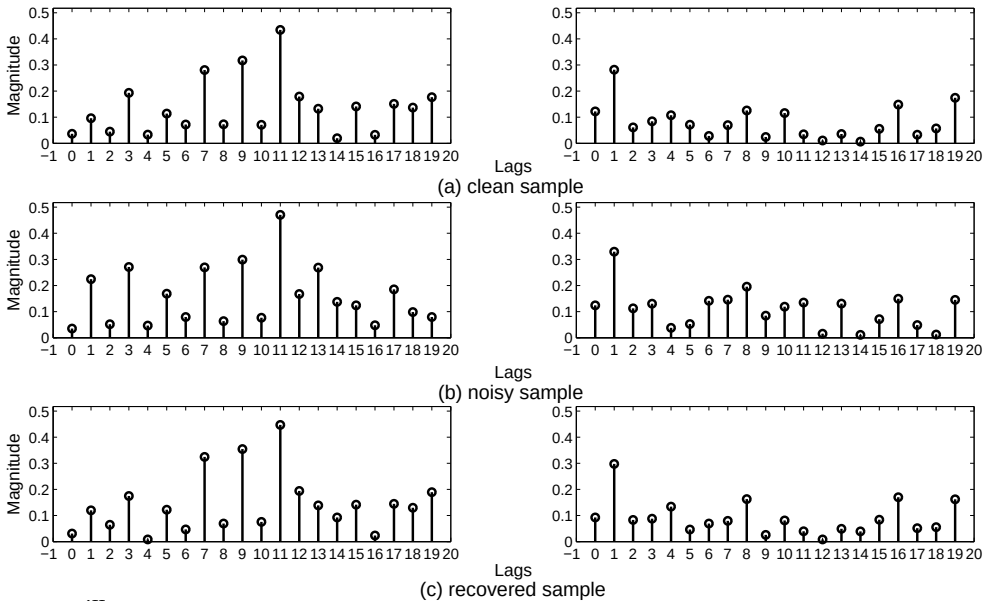

Fig. 3: (a) The clean impulse response signal. (b) The noisy impulse response signal which was obtained by adding 5 dB noise to the clean impulse response signal. (c) The impulse response recovered from the noisy version by the proposed method.

different noise levels. We can see that the error rate of the recovered signal is smaller than that of the noisy signal at all tested noise levels. This means that the recovered signal is much more similar to the clean signal as compared with the noisy signal.

The impulse response at lag ℓ can be expressed as the coefficient matrix of $\mathbf{A}(\ell)z^{-\ell}$. As we use polynomial matrix to simulate impulse responses, we carried out another experiment to evaluate the recovery accuracy of the proposed method at different time lags with the same noise level, SNR = 5 dB. In this

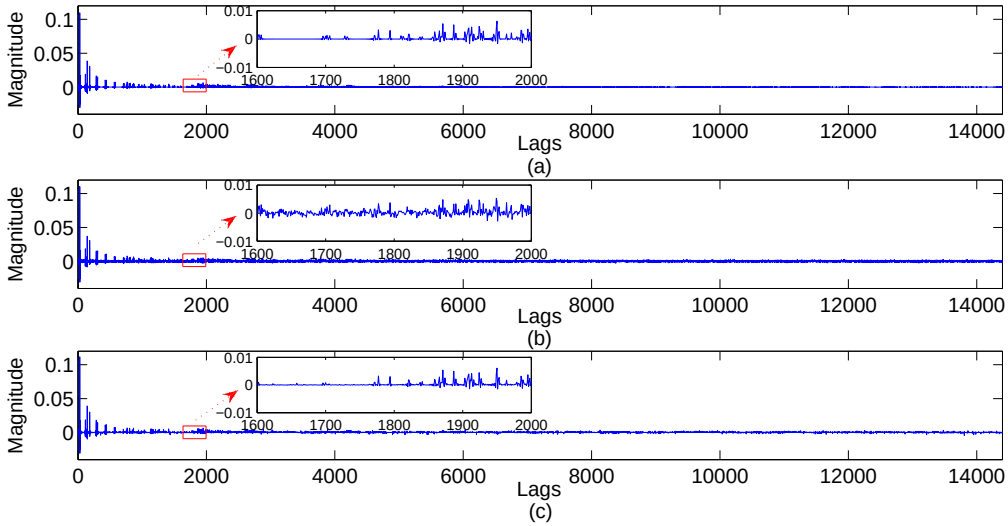

Fig. 4: (a) The clean acoustic impulse response signal. (b) The noisy acoustic impulse signal which was obtained by adding 5 dB white Gaussian noise to clean acoustic impulse signal. (c) The recovered acoustic impulse response signal.

experiment, the training data was constructed as a 5×10000 polynomial matrix. As the polynomial dictionary has to be overcomplete, the dictionary was designed to be a 5×200 polynomial matrix. The test data was a 5×100 polynomial matrix, and the sparsity was set to be 3. Figure 2 shows the error rates at different time lags of the simulated impulse response. By comparing the error rates E_n and E_r, it can be seen that the error rate of recovered signal is also smaller than that of the noisy signal. Therefore, our proposed method can recover noisy signal at different time lags.

An illustration of the impulse responses is provided in Figure 3, which shows two elements of the polynomial matrix (without noise), its corresponding noise added matrix and recovered matrix, where the lags of these polynomial matrices are 20. Compared with the clean signal, we can see from Figure 3 that our proposed method can reconstruct the polynomial matrix very well from the noisy samples. This figure shows the denoising ability of the proposed method for the reconstruction of the impulse responses from noisy measurements.

Experiments on Impulse Responses Generated by Image Room We conducted another experiment with a noisy impulse response signal generated

by a room image model [2]. The clean impulse responses generated by the room image model were used as training data to train a polynomial dictionary. A noisy test signal was generated by adding noise at SNR = 5 dB. The proposed algorithm was used to recover the clean impulse responses. It can be seen from Figure 4 that the recovered acoustic impulse is very similar to the clean one.

5 Conclusion

We have introduced a method for polynomial dictionary learning based on K-SVD algorithm. This provides a way for learning a dictionary of impulse responses for describing a room. Experiments on both synthetic and simulated room impulse responses show that the proposed dictionary learning method can be used for recovery of impulse responses corrupted by noise. The proposed method has the potential for speech dereverberation which could be achieved by controlling the sparsity of the representation coefficient matrix. In addition, the current method is based on the coefficients of the polynomial matrix, one could directly calculate the dictionary matrix based on the use of a polynomial SVD method [4]. These constitute our future work.

6 Acknowledgement

The work was conducted when J. Guan was visiting the University of Surrey, and supported in part by Shenzhen Applied Technology Engineering Laboratory for Internet Multimedia Application under Grants Shenzhen Development and Reform Commission, China (Grant Number 2012720).

References

1. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing 54(11), 4311–4322 (2006)
2. Allen, J.B., Berkley, D.A.: Image method for efficiently simulating small-room acoustics. The Journal of the Acoustical Society of America 65(4), 943–950 (1979)
3. Dai, W., Xu, T., Wang, W.: Simultaneous codeword optimization (simco) for dictionary update and learning. IEEE Transactions on Signal Processing 60(12), 6340–6353 (2012)
4. Foster, J.A., McWhirter, J.G., Davies, M.R., Chambers, J.A.: An algorithm for calculating the QR and singular value decompositions of polynomial matrices. IEEE Transactions on Signal Processing 58(3), 1263–1274 (2010)
5. Kreutz-Delgado, K., Murray, J.F., Rao, B.D., Engan, K., Lee, T.W., Sejnowski, T.J.: Dictionary learning algorithms for sparse representation. Neural Computation 15(2), 349–396 (2003)
6. Rota, L., Comon, P., Icart, S.: Blind MIMO paraunitary equalizer. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2003. vol. 4, pp. 285–288. IEEE (2003)
7. Saramäki, T., Bregovic, R.: Multirate systems and filter banks. Multirate Systems: Design and Applications 2, 27–85 (2001)