A NOVEL TIME-FREQUENCY FEATURE EXTRACTION ALGORITHM BASED ON DICTIONARY LEARNING

*Jefferson Medel, Andreas Savakis, Behnaz Ghoraani**
Department of Biomedical Engineering
Rochester Institute of Technology

ABSTRACT

Time-frequency (TF) representations have been widely used over the past decade to characterize the non-stationary content of signals in the joint time and frequency domain. Although a number of effective TF analysis methods based on wavelet or Gabor transform have been developed, these methods use pre-determined basis functions and still require feature extraction methods to reduce redundancy and preserve important TF information related to the application of interest. This paper explores a novel TF feature extraction algorithm using a modified dictionary learning approach. The proposed algorithm is developed to modify learned dictionaries and derive TF features unique to each class. It emulates the way joint dictionary learning algorithms use common dictionaries to promote discrimination between the data from different classes, thereby allowing for an improved analysis of complex and multitask data. The proposed method indicated a significant performance in identification of the discriminant vs. common structures of the TF data.

Index Terms— non-stationary, time-frequency, feature extraction, dictionary learning

1. INTRODUCTION

Time-frequency (TF) representation [1,2] has found wide use in many challenging signal processing tasks including classification, interference rejection and retrieval [3-7]. Furthermore, TF analysis offers a framework through which we can understand the underlying processes of complex, nonlinear and nonstationary systems. Developing effective feature extraction tools for modeling the TF representation is important for reducing dimensionality and redundancy, and obtaining the essential TF structure of the observed data that is necessary for understanding the data generation mechanism.

In this paper, we propose a new TF feature extraction algorithm based on dictionary learning that can handle challenging scenarios where the data is multi-task with overlapping classes and/or common structures. The proposed method obtains the TF dictionaries from each class and then modifies the class dictionaries to generate discriminant TF dictionaries for each signal category to promote separability, as well as a common dictionary to promote approximation. This is extremely important, as it provides a deep connection to the behavior of real world signals, which are a nonlinear combination of overlapping tasks with some common baseline structure. The modified discriminant and class-specific dictionaries are designed to represent the TF features that are unique to each class.

* Corresponding author. Email: bghoraani@ieee.org

1.1. PRIOR WORK

Although a number of effective TF analysis methods based on wavelet or Gabor transform [2] have been developed, these methods use pre-determined basis functions. Furthermore, they still require feature extraction methods to reduce redundancy and preserve important TF information related to the application of interest. Advances in TF analysis methods have led to the development of powerful techniques [4,8-10], which use matrix decomposition methods with different constraints such as independent component analysis (ICA) [12], principal component analysis (PCA) [11] or non-negative matrix factorization (NMF) [13] to adaptively decompose the TF data into TF basis components and coefficients. Instead of traditionally assuming the stationarity of the signal over short segments, these methods adaptively decompose the TF data into intervals with similar spectral characteristics. With this approach, relatively long durations of data are represented with a few basis components, which can be used as TF features because they represent spectral variations of the signal without any stationarity assumptions over predefined segments. However, the existing approaches are performed in a supervised fashion, meaning that the TF decomposition is performed separately for each category and then the TF features are used in a classifier to analyze the data. Hence, these approaches tend to fail to preserve the discriminative characteristics of each category and the common structures as separate TF bases, thereby, resulting in misrepresentation of the complex and multitask data.

2. MATERIALS AND METHODS

2.1. Dictionary Learning

The foundation of our approach stems from dictionary learning techniques for sparse representations [16], which have gained popularity in signal and image analysis during the past decade. We begin by considering a matrix $\mathbf{D}_{n\times m} = [\mathbf{D}_1, \mathbf{D}_2, \dots, \mathbf{D}_p]$ representing an over-complete dictionary of m samples [14-16], each of n -dimensions, drawn from p separate classes. Given a test sample $\boldsymbol{x}$, which belongs in the

qth class, its linear representation is obtained as $\boldsymbol{x} = \mathbf{D}\boldsymbol{a}$, where $\boldsymbol{a} \in R^n$ is a sparse coefficient vector whose entries are mostly zero, except for those associated with the qth class. We can obtain the sparse solution as follows:

$$\hat{\boldsymbol{a}} \triangleq \arg\min \frac{1}{2}\|\mathbf{D}\boldsymbol{a} - \boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{a}\|_1 \qquad (1)$$

where $\|\boldsymbol{a}\|_1 = \sum_i |\boldsymbol{a}|$ and the ℓ^1-norm minimization approach promotes sparse solutions and can be reformulated as a convex linear programming optimization method. λ is a regularization parameter which is can be adjusted to achieve sparser solutions. An applied penalty through the regularization term enforces sparsity and can be used for feature selection, excellent classification, and more interpretable solutions. Orthogonal Matching Pursuit (OMP) [17] is a popular method for solving for the sparse coefficients. Given the sparse coefficient vector $\hat{\boldsymbol{a}}$, minimum reconstruction error can be used to classify a test sample to class p.

Classical dictionary learning techniques for sparse representation consider a finite training set of signals and optimize the following empirical cost function to build an overcomplete dictionary to approximate the data in a sparse fashion.

$$\mathrm{D}^* \triangleq \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\boldsymbol{x_i} - \mathbf{D}\boldsymbol{a}\|_2^2 + \lambda\|\boldsymbol{a}\|_1 \qquad (2)$$

K-SVD [16] is an iterative dictionary learning method, where at each iteration, training samples are first sparsely coded using the current dictionary estimate, and then dictionary elements are updated one at a time while keeping others fixed. Each new dictionary element is a linear combination of training samples. Rubinstein *et al.* [18] implemented an efficient implementation of K-SVD using Batch OMP.

In this paper, we use non-negative KSVD (NN-KSVD) [19] for learning the initial class-specific dictionaries. NN-KSVD is our preferred method due to the way it allows signals in the TF domain to decompose into additive models of non-negative atoms, making them a more accurate representation of the non-negative data. Enforcing a non-negative constraint upon both the dictionary and the resulting coefficients also forces the dictionary elements to become sparser due to its inability to subtract, resulting in a more natural decomposition.

2.1. The Proposed TF Feature Extraction Algorithm

The proposed method is inspired by the joint dictionary learning algorithm proposed in [20] to obtain a common dictionary for images. The TF domain of a signal can be considered as an image where the intensity at each pixel is the energy of the signal at the corresponding time and frequency. In image analysis, the image data is often divided into smaller patches for further processing since only the intensity of a pixel matters; however, the same cannot be done for energy in different patches of data in the TF image because the time and/or the frequency of the pixel also contains important information about the structure of the data. Therefore, appropriate methods need to be developed in order to extract relevant and important information from TF images.

The proposed method takes in learned dictionaries of p classes and returns a modified and "discriminant" dictionary of each class along with a "common" dictionary. The modified dictionaries retain only TF data unique to that class. Any portions of data that are shared with another class are extracted and placed in the common dictionary.

The training data consist of subsamples of size $j \times k$ of TF signals of size $j \times l$ representing unique classes. The data used to train the dictionaries must have the same range in frequency, as well as the same dimensionality. Having the same frequency range ensures that all of the data between dictionaries are compatible. Making sure that each dictionary is comprised of the same number of atoms, ensures that the dimensionality of each dictionary is uniform.

Algorithm 1 Unique Feature Extraction

Input: Dictionaries $\{\mathbf{D}_i\}$ of size $n \times m$, $i = 1,\ldots,z$ data subsample sizes j and k, section size s, threshold value T

1. Resize all **D** such that every dictionary is size $j \times (k*m)$
2. Make a copy $\{\mathbf{F}_i\}$ of all $\{\mathbf{D}_i\}$
3. **repeat** $\{\mathbf{D}_i\}$
4. **repeat** {Initialize $r = 1$}
5. Extract a s by $(k*m)$ section for all **D** beginning at row r.
6. Collapse the extracted section back into size equal to $(s*k)$ by m.
7. Find all unique atoms that have a correlation above T with at least one atom of another section.
8. Iterate through the corresponding section of $\{\mathbf{F}_i\}_{i=h}$ and set the value all found atoms to zero.
9. Pad the unique atoms with zeros to line up its frequencies with that of an atom from **D** and add it to dictionary **C.**
10. Increment r
11. **until** r is equal to n-s
12. **for** $i = 1,\ldots,z$

Output: Discriminant Feature Dictionaries **F**, Common Feature Dictionary **C**

The details of the method are outlined in **Algorithm 1**. The algorithm filters any non-unique features from each class dictionary by checking how closely correlated a feature is with each feature in every other dictionary. This results in the modification of the original dictionaries such that the modified dictionaries only contain features unique to their own class. All non-unique features found are extracted and placed into a "common" dictionary of shared features. Step 5 is especially important because it defines the difference between image and TF data. In images, it is possible to compare any image patch with any other image patch. This cannot be done for TF signals because TF domain holds information on both time and frequency in addition to energy, making its position matter. This prevents a valid comparison of data when neither the time nor frequency is the same due to the lack of reference. Step 5 circum-

vents this issue by regulating the data such only data of similar frequencies are compared.

The parameter selection is application dependent and has to be tuned empirically. However, the size of the data and number of atoms can be tuned before implementing the algorithm through use of NN-KSVD reconstruction and classification errors. Defining the optimal size of an extracted section, as defined in step 4, can only be done through the algorithm though. The size of the section is particularly important because when it is combined with the subsampled training data, it provides a partitioned sample similar to that of an image patch that would be otherwise impossible to make. Keeping track of and regulating information by frequency allows for unique TF features to be found within a class regardless of the time or frequency.

3. RESULTS

3.1. Experimental Setup

A non-stationary synthetic dataset (see **Figure 1**) inspired by previous work in the literature [21] was generated for testing. The dataset is comprised of four classes, and several test signals. Each generated signal is multiplied by a Gaussian envelope, allowing for signals to be concatenated without discontinuities in frequency. The spectrogram of each signal is generated in MATLAB with a window of 128, a non-overlap of 125, and a cyclical frequency of 128. A total of two hundred signals are generated for each class. A total of one hundred fifty windows, each comprised of thirty columns, are then randomly selected and extracted from each signal. This resulted in a total of 30,000 samples used as training data.

Test signals were generated by concatenating relevant Class ***x***, Class ***y***, Class ***z***, and Class ***xy*** signals, as well as relevant combinations of pairs of classes. The test signals are shifted in the time domain to make sure that the algorithm is time-shift invariant. A window size of thirty columns and dictionary size of one hundred thirty atoms is found through cross-validation while training dictionaries to find reconstructions of the test signals with the lowest error.

The trained dictionaries generated for each class are used as inputs to test the algorithm. The algorithm returns a "discriminant" dictionary for each class, along with a "common" dictionary containing all non-unique features. The new dictionaries returned by the algorithm are then concatenated to form a single combined dictionary and used to linearly decompose the test signal into a sparse linear representation of each signal through the orthogonal matching pursuit algorithm [22]. The linear decomposition of the test signal ensures that the sparse representation is organized such that each row corresponds to a single atom, thus representing a single class. Relevant portions of the sparse representation of the signal are then used to reconstruct the test signal by class. The reconstructed signals by class should contain features unique to only that class.

The experiment is repeated under three different conditions. In the first experiment, two dictionaries are the input to the algorithm, trained by class ***x*** and class ***y*** respectively. In the second and third experiments, three dictionaries are the input to the algorithm, trained by class ***x***, class ***y***, class ***z***, and class ***x***, class ***y***, class ***xy,*** respectively.

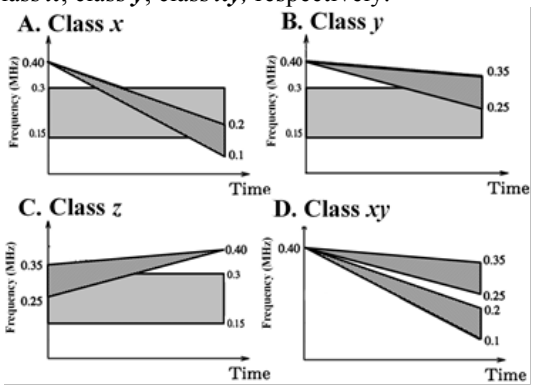

Figure 1. Synthetic Dataset. The TF structures of Class ***x*** (**A**), Class ***y*** (**B**), Class ***z*** (**C**), and Class ***xy*** (**D**) are shown in this figure. Each class consists of two distinct signals. Class ***x*** and ***y*** contain a uniformly distributed tone between 0.15 and 0.3Hz, and a linear chirp signal starting at 0.4Hz and ending at a random frequency uniformly distributed between 0.1 and 0.2Hz for class ***x***, and between 0.25 and 0.25Hz for class ***y***. Class ***z*** contains the same uniformly distributed tone, with a chirp signal beginning at a random frequency uniformly distributed between 0.25 and 0.35Hz and ending at 0.4Hz. Class ***xy*** is comprised of one linear chirp signals from class *x* and one from class ***y***.

3.2. Two Class Data with One Common Feature

The unique features of class ***x*** and class ***y***, characterized by a line with a steep slope and a more horizontal slope, respectively, are successfully extracted through a partial reconstruction of the signal using the modified class ***x*** and class ***y*** dictionaries (see **Figure 2**). The horizontal uniformly distributed tone between 0.15 and 0.3Hz common to both classes is successfully separated from the unique features of both classes and put into the "common" feature dictionary, allowing for a partial reconstruction of only the non-unique features. While the reconstructions are not perfect representations of the features represented by each dictionary, they are significantly improved compared to those of unmodified dictionaries.

3.3 Three Class Data with One Common Feature

The proposed algorithm is able to modify more than two dictionaries, such that the partial reconstructions using individual dictionaries with derived TF features are unique to the class it was representing (see **Figure 3**). The most successful unique feature extraction occurred for classes ***x*** and ***z***. While the reconstruction of features unique to class ***y*** is not wrong, there are imprints of the feature present in the partial reconstruction using the Common Feature dictionary. This can be attributed to the fact that because the class can be attributed to the fact that because the class ***y*** feature has a smaller slope, it is sometimes confused with the horizontal common features.

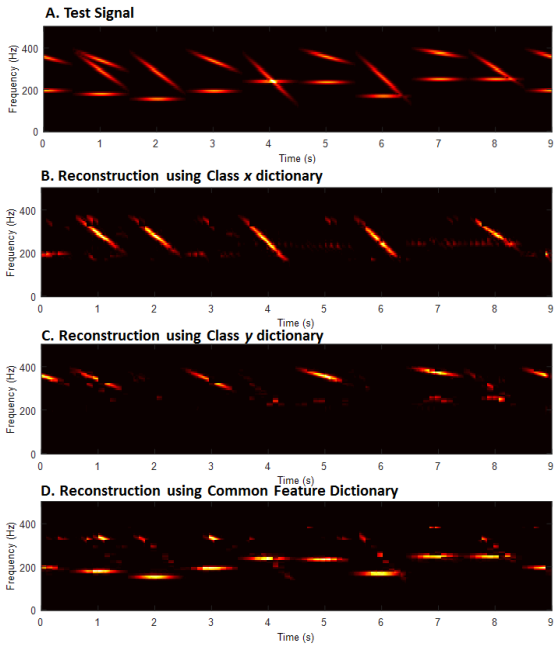

Figure 2. Reconstruction of Unique Features in classes x and y. A sparse representation of the test signal (**A**), comprised of a mix of class ***x*** and ***y*** signals, was used in partial reconstructions with each modified dictionary to obtain TF features unique to class ***x*** (**B**), class ***y*** (**C**), neither (**D**).

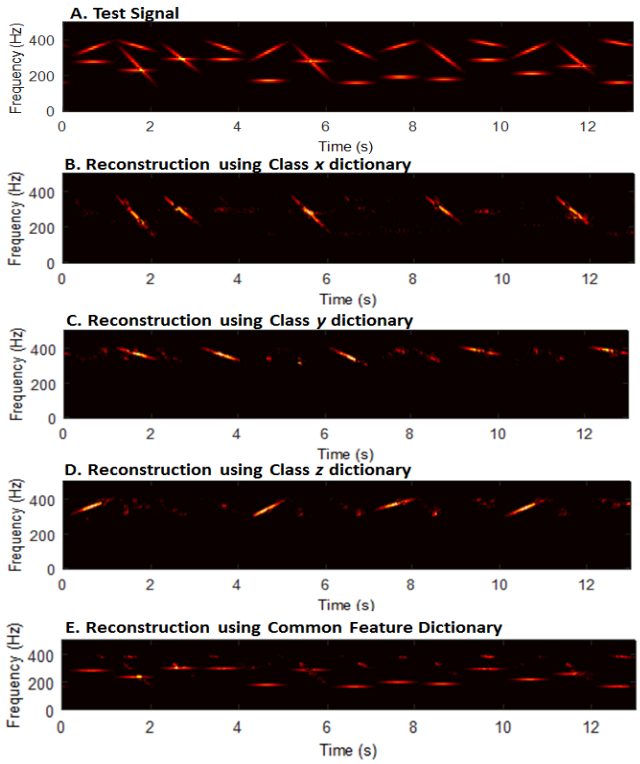

Figure 3. Extraction of Unique Features Between Class ***x***, ***y*** and ***z***. A sparse representation of the test signal (**A**), comprised of a mix of ***x, y,*** and ***z*** signals, was used in partial reconstructions with each modified dictionary to obtain features unique to class ***x*** (**B**), class ***y*** (**C**), class ***z*** (**D**), and neither (**E**).

3.4 Three Class Data with No Unique Features

Classes ***x, y,*** and ***xy*** all share at least one feature with one another. Consequently, the algorithm should not find any features unique to a class, ideally modifying the class dictionaries to be empty and placing everything within the "common" feature dictionary. While the results are not perfect (see **Figure 4**), they are successful and can be useful for classification. The reconstruction of features unique to classes ***x*** and ***y*** through their respective dictionaries yield no recognizable patterns or features, while the reconstruction of class ***xy*** though its dictionary yielded only light partial imprints of the class. The reconstruction of the "common" feature dictionary results in a nearly complete reconstruction of the test signal, showing that the classes do not contain unique features.

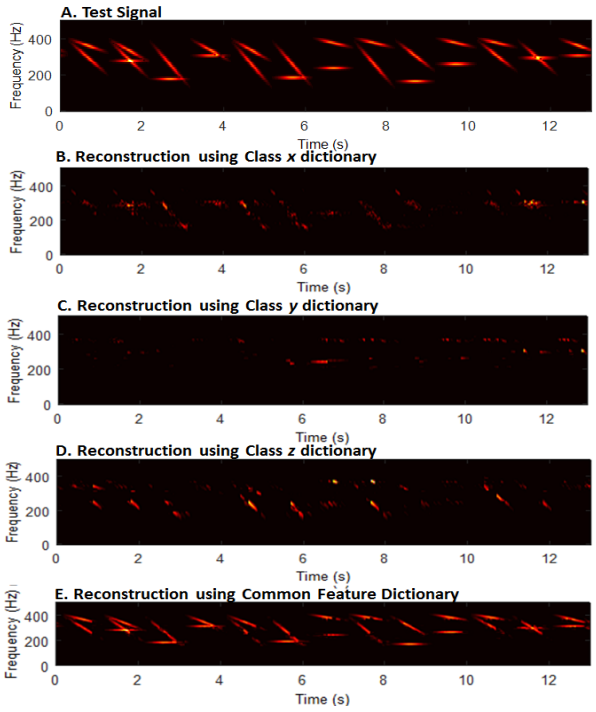

Figure 4. Extraction of Unique Features Between Class ***x***, ***y*** and ***xy***. A test signal (**A**), comprised of a mix of ***x, y,*** and ***xy*** signals and the estimated representation of class ***x*** (**B**), class ***y*** (**C**), class ***xy*** (**D**), and neither (**E**).

4. CONCLUSION

In this paper, a novel TF feature extraction algorithm was developed to extract features unique to each class. The application of the proposed algorithm on a synthetic dataset with two or three overlapping classes demonstrated its effectiveness for a better analysis of the characteristics that comprise a class. This method can be modified to organize the common information in several dictionaries to highlight what classes the shared features represent, or combined with NMF to improve the accuracy of TF feature extraction. Furthermore, since this algorithm offers better discrimination between class dictionaries, it can be used improve the classification performance for TF feature-based applications.

5. REFERENCES

[1] S. G. Mallat, Z. Zhang, "Matching pursuits with time-frequency dictionaries," IEEE Transactions on signal processing, 41, 3397–3415, 1993.

[2] L. Cohen, "Time-frequency analysis" Vol. 778. Englewood Cliffs, NJ. Prentice Hall PTR, 1995.

[3] B. Ghoraani, S. Krishnan, "A Joint Time-Frequency and Matrix Decomposition Feature Extraction Methodology for Pathological Voice Classification", the EURASIP Journal on Advances in Signal Processing, Article ID 928974, 11 pages, doi:10.1155/2009/928974, 2009.

[4] B. Ghoraani, S. Krishnan, "Time-Frequency Matrix Feature Extraction and Classification of Environmental Audio Signals", the IEEE Transactions on Audio, Speech and Language Processing, 19 (7), 2197 – 2209, 2011.

[5] B. Ghoraani, S. Krishnan, R. J. Selvaraj, V. S. Chauhan, "T Wave Alternans Evaluation Using Adaptive Time-Frequency Signal Analysis and Non-negative Matrix Factorization", Medical Engineering and Physics, 33(6), 700-711, 2011.

[6] D. Jiang, Z. Xu, Z. Chen, Y. Han, H. Xu. "Joint time–frequency sparse estimation of large-scale network traffic". Computer Networks, 55(15), 3533-3547, 2011.

[7] A. Gramfort, D. Strohmeier, J. Haueisen, M. S. Hämäläinen, M. Kowalski, "Time-frequency mixed-norm estimates: sparse M/EEG imaging with non-stationary source activations". NeuroImage, 70, 410-422, 2013.

[8] C. Févotte, N. Bertin, J. L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis". Neural computation, 21(3), 793-830, 2009.

[9] R. Hennequin, B. David, R. Badeau, "Score informed audio source separation using a parametric model of non-negative spectrogram". In Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2011.

[10] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria". IEEE Transactions on Audio, Speech, and Language Processing,15(3), 1066-1074, 2007.

[11] I. Jolliffe, "Principal component analysis". John Wiley & Sons, Ltd, 2005.

[12] J. Bell, T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Comput., 7(6), 1129–1159, 1995.

[13] D. Lee, H. Seung, "Algorithms for non-negative matrix factorization," in Proc. 2000 Conf. Adv. in Neural Inf. Process. Syst. 13, 556–562, 2001.

[14] J. A. Tropp, A.C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit." IEEE Transactions on Information Theory, 53(12), 4655-4666, 2007.

[15] D. L. Donoho, M. Elad, V. Temlyakov, "Stable recovery of sparse overcomplete representations," IEEE Transactions on Information Theory, 2005.

[16] M. Aharon, M. Elad, A.M. Bruckstein. KSVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation." *Signal Processing, IEEE Transactions on* 54.11 (2006): 4311-4322.

[17] D. L. Donoho, Y. Tsaig, "Fast Solution of L1-norm Minimization Problems When the Solution May Be Sparse," Stanford CA, 94305, Department of Statistics, Stanford University, 2006.

[18] M. Aharon, M. Elad, A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," IEEE Trans. Signal Process., vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[19] M. Aharon, M. Elad, A. Bruckstein. "K-SVD and its non-negative variant for dictionary design." Optics & Photonics 2005. International Society for Optics and Photonics, 591411, 2005.

[20] N. Zhou, F. Jianping "Jointly learning visually correlated dictionaries for large-scale visual recognition applications." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36.4 (2014): 715-730.

[21] M. Davy, C. Doncarli, G. F. Boudreaux-Bartels, "Improved optimization of time-frequency based signal classifiers". IEEE Signal Process. Lett. 8, 52–57, 2001.

[22] R. Rubinstein , M. Zibulevsky M. Elad "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit", 40(8): 1-15, 2008.