TOWARDS TIME-DOMAIN PIANO MUSIC TRANSCRIPTION USING K-SVD

Zachary Neveu

Technical University of Denmark

ABSTRACT

This work presents a method for transcribing piano music in the time domain using k-singular value decomposition (KSVD) given existing note onsets. The method presented allows for learning from small datasets and offers the potential to generalize to instruments with widely varying spectral characteristics and non-western musical scales.

***Index Terms*—** K-SVD, Factor Models, Music
code at: https://github.com/zacharyneveu/MLSP_Project

1. INTRODUCTION

Music, in general, has a well defined structure that can be represented in low-dimensional forms such as sheet music or musical instrument digital interface (MIDI) scores. In these compressed forms, the salient features of music (note pitches, velocities, and timing) are stored directly, and can be searched and manipulated easily. Musical performance often deviates in interesting ways from written music, however live performances are most often recorded in audio form and require significant human effort to analyze or manipulate. Transforming music from a high-dimensional audio signal into a sparse symbolic form can be performed algorithmically which can save time and effort.

2. BACKGROUND

Traditional signal processing approaches can perform transcription for certain musical instruments and styles with a moderate level of success [1] . These approaches generally utilize the fact that a musical note is written as the lowest frequency (F_0) that occurs when this note is played. In the subtask of melody transcription, it is known that only one note can occur simultaneously, so simply finding the lowest prominent frequency in the time-frequency domain can often be adequate to transcribe a sample. In polyphonic transcription, however, multiple notes can be played at once. This problem can be approached well in a statistical context using maximum a-priori (MAP) estimation to determine the most likely combination of notes occurring given a set of frequencies [2]. Another approach to polyphonic transcription is to factor the magnitude spectrogram of the signal. Because of the nature of the short time Fourier transform (STFT), the magnitude of each time-frequency bin must be non-negative. In order to exploit this, non-negative matrix factorization (NMF) has been applied for spectrogram factorization by [3]. For piano music, another known constraint is that representations are sparse in the frequency domain. To utilize this, [4] used non-negative k-singular value decomposition (NKSVD) in the spectrogram domain to constrain the sparsity of solutions along with the positivity constraint. More recently, neural networks have been applied to the polyphonic transcription task, most notably in [5]. Neural network approaches are able to benefit from learning musical structure, grasping the nonlinear relationships between notes over time, making them the most performant approach for polyphonic transcription. Neural architectures suffer from the lack of large-scale labeled data in this area, however, because human transcription requires skilled and intensive labor, and performances are rarely recorded in MIDI form.

3. PROBLEM FORMULATION

For the purposes of this paper, it is assumed that a method for onset detection such as the one presented in [6] is available and performs well. The input for the problem is then time-domain frames of music each of which contains one or more onsets located exclusively at the beginning. The goal is to factor a set of frames, X, into an approximate linear combination, Z, of a dictionary of notes, $\mathbf{D}$, as expressed in equation (1). The dimensionality of the variables are $X \in R^{n\times N}$, $D \in R^{n\times K}$, and $Z \in R^{K\times N}$ where N is the number of audio frames, n is the length of a frame, and K is the number of atoms in the dictionary. The constraint is imposed that the number of non-zero elements in $Z \leq k$ where $k \ll K$. For our specific purposes K was chosen to be 88 because a piano has 88 possible notes, and k was chosen to be 10 as the average pianist has 10 fingers. Solving for $\mathbf{D}$ and Z exactly is an NP-hard problem, so for practical purposes results must be estimated.

$$X \approx \mathbf{D}Z \quad s.t. \; \|Z\|_0 \leq k \tag{1}$$

4. DATA

A subset of 2000 frames from the MAESTRO dataset[7] is used to fit the dictionary. This data consists of audio files of piano music, alongside MIDI files which contain the times, frequencies, and amplitudes of each note played. These audio files are oversampled to $44,100\,Hz$ to cover the full range of human hearing, however this is generally unnecessary for music transcription (the highest piano note has an F_0 of $4186\,Hz$), so the audio files are downsampled by a factor of 4 to $11,025\,Hz$. Using the MIDI annotations, audio files are then split at note onsets. Each frame is then zero-padded or cropped so that the length of all frames is identical and of a tractable size of 512 samples. These frames were then concatenated into a 2000×512 matrix suitable for training the dictionary.

5. ALGORITHM

The KSVD algorithm was originally introduced in [8] and provides a way to learn an over complete dictionary with which to represent signals with a given sparsity. Listing 1 shows an overview of the KSVD algorithm for signal X, dictionary D and a given maximum number of iterations. Essentially, the algorithm is an iterative process with two parts. First a sparse combination of dictionary atoms is found that best fit the signal using the orthogonal matching pursuit (OMP) algorithm seen in Listing 1 [9]. Next, the dictionary D is updated one atom at a time so that the selected atom contributes as possible to the sparse representations of all data points which use it. These two steps are repeated either convergence or until a specified maximum number of iterations.

```
function KSVD(X,D,max_iter)
    # Initialization
    D = random(n,K)
    for j=1:max_iter
        # Sparse Coding
        Z = OMP(X, D, k)
        # Dictionary Update
        for k=1:K
            # create error matrix
            $E_k$ = X-(DZ-D[k]Z[k])
            $w_k$ = ones(N, 1)
            $w_k$[k] = 0
            $E_k$ = $E_k$[wk,:]
            D[k], Z[k] = SVD($E_k$)
        end
    end
    return D, Z
end
```

Listing 1: K-SVD Algorithm [8]

```
function OMP(X, D, max_nonzeros)
    residual = X
    active_atoms = []
    for i=1:max_nonzeros
        atom_corrs = D'*residual
        atom_idx = argmax(abs.(atom_corrs))
        push(active_atoms, atom_idx)
        selected_atoms = D[:,active_atoms]
        z = least_squares(selected_atoms, X)
        residual = residual-selected_atoms*z
    end
    return_val = zeros(n_atoms)
    return_val[active_atoms] = z;
    return return_val
end
```

Listing 2: Orthogonal Matching Pursuit Algorithm [9]

In order to reconstruct frames after D is trained, OMP can be used without performing KSVD after. The task of fitting Z given X and D is NP-Hard. OMP is an iterative greedy algorithm that gives very good approximate results. The premise behind OMP is to select atoms that are most correlated to the parts of the input signal not yet represented by chosen atoms. First, the most correlated atom is chosen, and its weight is fit using least squares (LS). The weighted atom is then subtracted from the residual and the process is repeated. The benefit to using this approach over the original approach to matching pursuit introduced by [10] is that using OMP, the atoms are chosen to be orthogonal. This means that instead of choosing two atoms and two weights to represent an aspect of the signal, OMP will instead choose a larger weight, freeing up an atom to be used for another purpose.

6. RESULTS

To evaluate the results of KSVD, the learned dictionary atoms, as well as example signal reconstructions are examined. For the specific task at hand, it is known that piano music is sparse in the frequency domain, so the dictionary atoms should have learned meaningful frequency-domain content. To evaluate this, the fast Fourier transform (FFT) of each atom was calculated. Figure 1 shows the magnitudes of the FFT for each atom sorted by the frequency of maximum component. From figure 1, it is clear that KSVD has learned the primary frequencies of a variety of notes. It is important to note that the training data does not necessarily include every note on a piano, and this may be the reason that certain notes have multiple atoms while others do not have an atom. To evaluate the reconstructed data, several audio frames not seen in the training dataset are reconstructed using the learned dictionary and the OMP algorithm. Figure 2 shows a subset of one of these frames, approximated using 10 atoms, as well as all atoms from the dictionary.

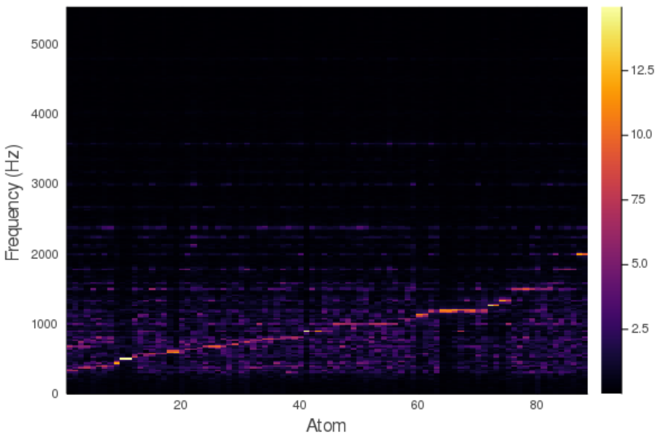

Fig. 1. Learned Frequencies of Dictionary Atoms

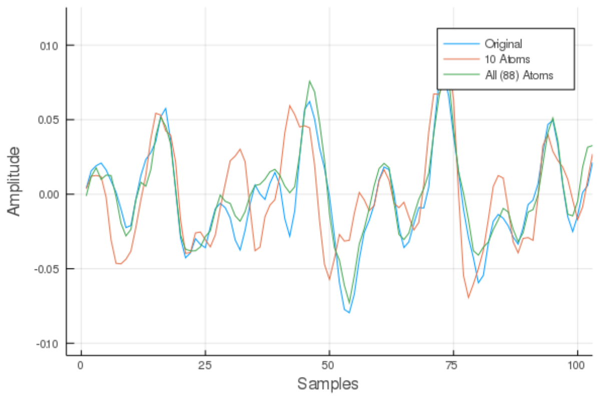

Fig. 2. Example of Reconstructed Signal with 10 and 88 Atoms

7. CONCLUSIONS

The conclusions of this work are that KSVD is capable of extracting notes from piano music in the time domain. This learning method can include a parameter specifying the maximum number of non-zero notes per frame, as well as the maximum number of possible notes to learn. These features make the technique generalizable to a large variety of instruments. One continuation of this work that could further improve performance would be to learn a complex, rather than real, vector Z. In the current scheme with a real value of Z_i, atom D_i contributes best only when it has a phase of either 0° or 180°. Making Z_i complex would allow D_i to contribute more when X is out of phase from the other training data. Another continuation of this work would be to pair the current algorithm with a note onset detection algorithm and measure the performance transcribing when there are some errors in the frame boundaries. Under these conditions, it would be possible to determine the extent to which this algorithm is feasible for real-world tasks.

8. REFERENCES

[1] Anssi Klapuri, *Signal Processing Methods for the Automatic Transcription of Music*, Ph.D. thesis, Tampere University of Technology, Tampere, 2004, OCLC: 76845615.

[2] Valentin Emiya, Roland Badeau, and Bertrand David, “Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2010.

[3] P. Smaragdis and J. C. Brown, “Non-negative matrix factorization for polyphonic music transcription,” 2003, vol. 2003-, pp. 177–180, IEEE.

[4] N. Bertin, R. Badeau, and G. Richard, “Blind Signal Decompositions for Automatic Transcription of Polyphonic Music: NMF and K-SVD on the Benchmark,” 2007, vol. 1, pp. I–65–I–68, IEEE.

[5] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck, “Onsets and Frames: Dual-Objective Piano Transcription,” *arXiv:1710.11153 [cs, eess, stat]*, June 2018.

[6] Sebastian Bock, Florian Krebs, and Markus Schedl, “EVALUATING THE ONLINE CAPABILITIES OF ONSET DETECTION METHODS,” p. 6.

[7] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck, “Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset,” *arXiv:1810.12247 [cs, eess, stat]*, Jan. 2019.

[8] Michal Aharon, Michael Elad, and Alfred M. Bruckstein, “K-SVD and its non-negative variant for dictionary design,” in *Optics & Photonics 2005*, Manos Papadakis, Andrew F. Laine, and Michael A. Unser, Eds., San Diego, California, USA, Aug. 2005, p. 591411.

[9] J. A. Tropp, “Greed is good: Algorithmic results for sparse approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[10] S.G. Mallat and Zhifeng Zhang, “Matching pursuits with time-frequency dictionaries,” *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec./1993.