Melody Extraction from Polyphonic Music

Rohin Garg
160583

Dr. Vipul Arora
Project Supervisor and Mentor

Report for EE392: Undergraduate Project 1

1 Introduction

Melody: A popular definition is that "the melody is the single (monophonic) pitch sequence that a listener might reproduce if asked to whistle or hum a piece of polyphonic music, and that a listener would recognize as being the essence of that music when heard in comparison."
This definition is open to interpretation, and is a very subjective one. Different listeners might hum different parts.
In a vocal centric audio, most will hum the vocal frequencies, but in instrumental pieces, different people may follow different instruments as the melody.
In practice, research has focused on single source predominant fundamental frequency estimation, that is melody is constrained to belong to a single sound source throughout the piece being analyzed, where this sound source is considered to be the most predominant instrument or voice in the mixture.

Melody Extraction: Given a musical audio, output a frequency value for every time instant representing the pitch of the dominant melodic line in the audio.

Melody Extraction from a polyphonic audio is a difficult task, in the sense that the term melody is subjective, and cannot be given a generalised mathematical definition for all music. But most of the methods till a few years back try to do so. A possible way to solve this problem is to use data driven methods. Not many people have tried out this approach. Most recent ones are Deep Salience by Rachel M. Bittner et. all. in 2016 [1] and Source Filter NMF with CRNN in 2017 by Dogac Basaran et. all [2]. These methods are able to match the existing state of the art results.
The first one is heavily data dependent. The second tries solve this problem by bringing in a mid-level representation of audio using Source Filter - Non Negative Matrix Factorisation (SF-NMF), instead of simply feeding the audio to the neural network.

I have tried to solve this problem through an intelligent mid-level representation of audio, improving upon SF-NMF, which also gives more control over the melody being extracted, solving the subjectivity problem to some extent.

1.1 Structure of the Report

This report is structured as follows:

1. Section 2 gives a brief overview of the current state of the art data driven model for melody extraction.
2. Section 3 explains the SF-NMF model, giving the math behind it and how it works. It also explains the way I have implemented it for my experiments.

3. Section 4 introduces the Melody specific Extended SF-NMF model, giving the intuition behind it, and what changes my model brings to the original, and it's advantages.
4. Section 5 gives the details of the various experiments performed, and gives a quantitative analysis for them.
5. Section 6 concludes this report, summarising the work done briefly, and states some ideas for future work.

2 Source Filter NMF with CRNN

They propose a Convolutional-Recurrent Neural Network (CRNN) model whose pre-training is based on the SF-NMF model proposed by Jean-Louis Durrieu et all. in 2011 [2]. They show that with NMF-based pretraining, one can achieve state-of-the-art results without requiring large training datasets or data augmentation methods, and using relatively simpler networks in terms of training parameters.

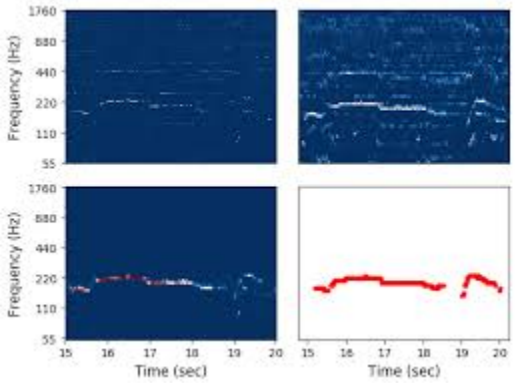


Figure 1: SF-CRNN Model [2]

2.1 Source Filter Model

The dominant melody in polyphonic audio is modeled using Non Negative Matrix Factorization (NMF):

$$\mathbf{S^X} = \mathbf{S}^{\phi} \odot \mathbf{S^{F_0}} + \mathbf{S^B}$$

$$\mathbf{S^X} = (\mathbf{W^\Gamma H^\Gamma H^\Phi}) \odot (\mathbf{W^{F_0} H^{F_0}}) + \mathbf{W^B H^B}$$

Here, $\boldsymbol{H^{F_0}}$ represents the pitch salience, which is independent from the timbre. $\boldsymbol{B}$ denotes the background audio spectrum.
Φ denotes the filter that catch the spectral envelope (related to timbre). Exactly how this factorisation works will be dealt in the section 3.

2.2 CRNN model

A combination of CNN and RNN is used to make a network that learns the melody from pitch salience.

The pitch salience, $\boldsymbol{H^{F_0}}$ is enhanced by the CNN stage. Since each CNN architecture only applies 2D linear filters and non-linear activations, the input structure is not lost through the layers of the network. This provides an advantage of interpretable hidden layer activations and leads to a new form of salience as output where each row still represents the activation of a fundamental frequency.

For the RNN, they use a single bidirectional Gated Recurrent Unit (BiGRU) layer to capture temporal relationships between F0s.
The final layer of the system is a classifier where one class represents the non-melody and the rest

of the 61 classes represent semitone fundamental frequencies between A1 and A6 (inclusive). The multiclass classification output is obtained with a single dense layer and softmax activations.

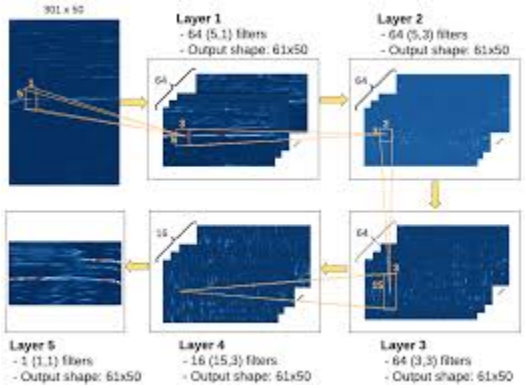


Figure 2: CRNN Model [2]

3 Source Filter NMF

Source Filter NMF first introduced in [3] is used to get a mid level representations of a polyphonic audio signal. It gives us salience representations which further facilitate various tasks, which traditional time-frequency domain representations like STFT cannot.

Consider the short time power spectrum of a polyphonic audio: $\mathbf{S^X}$.
This is assumed to be the sum of 2 independent spectrum:

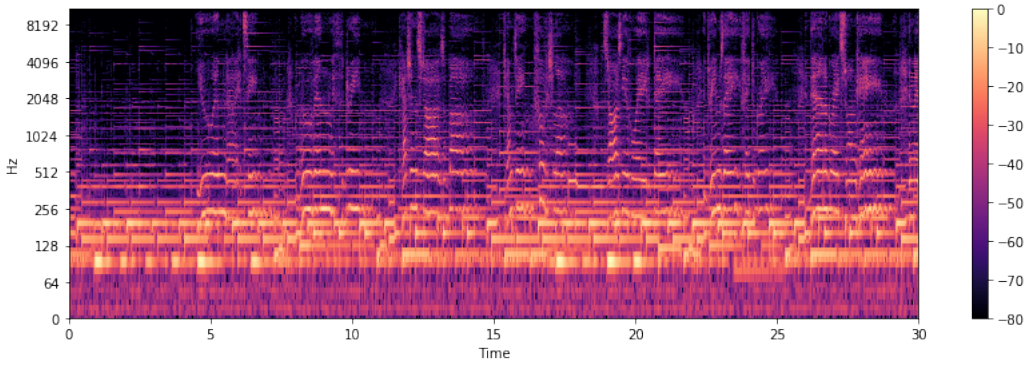


Figure 3: Power spectrum $\mathbf{S_X}$ in dB scale

$$\mathbf{S^X} = \mathbf{S^V} + \mathbf{S^B}$$

$\boldsymbol{V}$: the signal of interest, $\boldsymbol{B}$: residual signal.
Each Power spectrum is $F \times N$ matrix, F being the total frequency bins, N being total time bins.
Further,

$$\mathbf{S^V} = \mathbf{S^\Phi S^{F_0}}$$

Matrices $\mathbf{S^\Phi}$ and $\mathbf{S^{F_0}}$ capture different characteristics of the input signal of interest: F_0 recalls that the pitch information is included in the source part. Φ captures the timbre quality of the source, independent of pitch.

A simpler way to understand this is, for instance, when a singer sings an A4 (440 Hz) note, but sings different vowels, e.g., a $[a]$ at a frame n_1 and a $[e]$ at a frame n_2, then we would expect that

$\mathbf{s}_{\mathbf{n_1}}^{\mathbf{F_0}}$ and $\mathbf{s}_{\mathbf{n_2}}^{\mathbf{F_0}}$ to roughly contain comparable values, while the spectral envelopes $\mathbf{s}_{\mathbf{n_1}}^{\mathbf{\Phi}}$ and $\mathbf{s}_{\mathbf{n_2}}^{\mathbf{\Phi}}$ should be rather different and characteristic of the pronounced vowel. This model is therefore remotely related to a source/filter model.

3.1 Constructing the Model

A pitch salience representation with possibly concurrent notes(musical) is desired, so for each time bin n and frequency bin f :

$$s_{fn}^{F_0} = \sum_{u=1}^{U} h_{un} P_f(u)$$

$$h_{fn} >= 0$$

Here U is the total number of fundamental frequencies considered, and $P_f(u)$ a spectral shape centered at the fundamental frequency corresponding to index $u \in (1, U)$.
The original paper [3] uses glottal source model KLGLOTT88 as their spectral shapes. I have simply used Gaussian as a spectral shape centered at u (and it's harmonics), with suitable variance. The variance was chosen such that the gaussian overlaps with energy spread in the power spectrum.

Now, for each u, a fundamental frequency $F(u)$ is chosen, and the resulting combs generated from the spectral shapes are stored in $F \times U$ dictionary matrix $\boldsymbol{W}^{\boldsymbol{F_0}}$.
$F(u)$ varies with u logarithmically every U_{st} semitone from F_{min} to F_{max},

$$F(u) = 2^{\frac{u-1}{12U_{st}}} F_{min} \qquad \forall u \in [1, U]$$

where U_{st} is the number of frequencies per semitone.

The filter part aims at providing more flexibility to the model, adapting it to a variety of possible instances (recording conditions, velocity of the played notes, intonations for a voice, etc.). It is then decomposed into a linear combination of smooth filters $\Phi_k(f)$. Smoothness is ensured by generating them as a weighted sum of smooth spectrums:

$$s_{fn}^{\Phi} = \sum_{k}^{K} h_{kn}^{\Phi} \Phi_k(f) = \sum_{k}^{K} h_{kn}^{\Phi} (\sum_{p}^{P} h_{pk}^{\Gamma} \Gamma_p(f))$$

The smooth spectrum chosen is Hann functions.They make up the $F \times P$ matrix $\boldsymbol{W}^{\boldsymbol{\Gamma}}$.
So, now we have:

$$\mathbf{S^V} = (\mathbf{W^\Gamma H^\Gamma H^\Phi}) \odot (\mathbf{W^{F_0} H^{F_0}})$$

$$\mathbf{S^X} = (\mathbf{W^\Gamma H^\Gamma H^\Phi}) \odot (\mathbf{W^{F_0} H^{F_0}}) + \mathbf{W^B H^B}$$

where,

$$\mathbf{W^\Phi} = \mathbf{W^\Gamma H^\Gamma}$$

3.2 NMF algorithm

Unknown Parameters :

$$\Theta = \{H^\Gamma, H^{F_0}, H^\Phi, W^B, H^B\}$$

The estimation of parameters is based on the multiplicative update algorithms by D. D. Lee and H. S. Seung. in [4].
Θ is estimated such that the divergence between the power spectrum $\mathbf{S^{X,o}}$ and $\mathbf{S^X}$ is minimized. The divergence rule used can be generalised to β divergence:

$$d_\beta(a,b) = \begin{cases} a\frac{a^{\beta-1}-b^{\beta-1}}{\beta(\beta-1)} + b^{\beta-1}\frac{b-1}{\beta}, & \beta \in R^+ l;' \setminus \{0,1\} \\ a\log\frac{a}{b} - a + b, & \beta = 1 \\ \frac{a}{b} - \log\frac{a}{b} - 1, & \beta = 0 \end{cases}$$

The update equations for Θ, using β divergence are:

$$\mathbf{H^{F_0}} \leftarrow \mathbf{H^{F_0}} \odot \frac{(\mathbf{W^{F_0}})^\top (\mathbf{S^{\Phi}} \odot (\mathbf{S^{X}})^{\cdot(\beta-\mathbf{2})} \odot \mathbf{S^{X_0}})}{(\mathbf{W^{F_0}})^\top (\mathbf{S^{\Phi}} \odot (\mathbf{S^{X}})^{\cdot(\beta-\mathbf{1})})}$$

$$\mathbf{H^{\Phi}} \leftarrow \mathbf{H^{\Phi}} \odot \frac{(\mathbf{W^{\Phi}})^\top (\mathbf{S^{F_0}} \odot (\mathbf{S^{X}})^{\cdot(\beta-\mathbf{2})} \odot \mathbf{S^{X_0}})}{(\mathbf{W^{\Phi}})^\top (\mathbf{S^{F_0}} \odot (\mathbf{S^{X}})^{\cdot(\beta-\mathbf{1})})}$$

$$\mathbf{H^{\Gamma}} \leftarrow \mathbf{H^{\Gamma}} \odot \frac{(\mathbf{W^{\Gamma}})^\top (\mathbf{S^{F_0}} \odot (\mathbf{S^{X}})^{\cdot(\beta-\mathbf{2})} \odot \mathbf{S^{X_0}})(\mathbf{H^{\Phi}})^\top}{(\mathbf{W^{\Gamma}})^\top (\mathbf{S^{F_0}} \odot (\mathbf{S^{X}})^{\cdot(\beta-\mathbf{1})})(\mathbf{H^{\Phi}})^\top}$$

$$\mathbf{H^{B}} \leftarrow \mathbf{H^{B}} \odot \frac{(\mathbf{W^{B}})^\top ((\mathbf{S^{X}})^{\cdot(\beta-\mathbf{2})} \odot \mathbf{S^{X_0}})}{(\mathbf{W^{B}})^\top (\mathbf{S^{X}})^{\cdot(\beta-\mathbf{1})}}$$

$$\mathbf{W^{B}} \leftarrow \mathbf{W^{B}} \odot \frac{((\mathbf{S^{X}})^{\cdot(\beta-\mathbf{2})} \odot \mathbf{S^{X_0}})(\mathbf{H^{B}})^\top}{(\mathbf{S^{X}})^{\cdot(\beta-\mathbf{1})}(\mathbf{H^{B}})^\top}$$

The Divergence used is the ItakuraSaito (IS) divergence ($\beta = 0$).

3.3 My Implementation

I simplified the model by using Gaussian spectral shapes for dictionary matrix $\mathbf{W^{F_0}}$ centered at $F(u)$ (and it's harmonics), with suitable variance. The variance was chosen such that the Gaussian overlaps with energy spread in the power spectrum.
Also, for a fundamental frequency f_0, the while calculating the dictionary element for it's harmonics, the elements were scaled down by the square of the harmonic number.

$$\therefore \mathbf{W^{F_0}}[f, u] = \sum_{i=1}^{i*F(u)<F} \frac{1}{i^2} \exp - \frac{(f - i * F(u))^2}{\sigma^2}$$

As for the rest, I tried to stay as close to the actual model.

4 Melody Specific SF-NMF Extension

4.1 Intuition

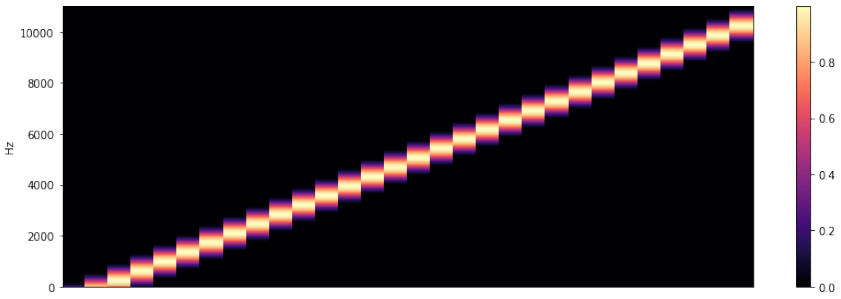
I analysed the pitch salience representation $\mathbf{H^{F_0}}$ obtained and the $\mathbf{S^{F_0}}$ estimated by the above model. The SF-NMF model as described does not give any freedom in defining what the user perceives as 'melody'.
This does not completely solve the problem of melody extraction, and as a mid-level representation, which is further used to extract melody via different methods, a neural network in this case, it should provide more flexibility with the pitch salience.

Also, the current melody extraction method does not use the timbre features estimated in $\mathbf{S^{\Phi}}$. These features can play an important role by giving more weight to the vocal melody over instrumental melody, or vice-versa.
They can act as a fingerprint for a particular singer or an instrument.

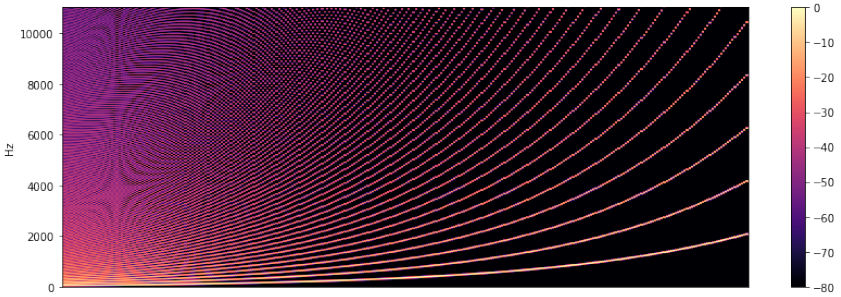
My contribution is based on the idea of controlling the timbre features and using them to estimate the pitch salience representation, using ideas from transfer learning.

4.2 Pre-Training the Timbre Features

For a polyphonic audio signal $\mathbf{X}$, let us denote the vocals by $\mathbf{V}$. We will consider the vocals to be the desired melody right now. This can be replaced by another instrument for non vocal centric music.
We need monophonic audio containing just the vocal samples of the same vocalist. 5 to 10 minutes



(a) $\mathbf{W}^{\mathbf{\Gamma}}$ using Hann functions



(b) $\mathbf{W}^{\mathbf{Fo}}$ using Gaussians

Figure 4: Fixed Bases Matrices for NMF

of samples should be sufficient, depending on the quality of samples available. For my experiments, I used the **MIR-1K** [5] and **Bach10** [6] dataset for this purpose.

The SF-NMF model:

$$\mathbf{S}^{\mathbf{X}} = (\mathbf{W}^{\mathbf{\Gamma}}\mathbf{H}_{\mathbf{v}}^{\mathbf{\Gamma}}\mathbf{H}^{\mathbf{\Phi}}) \odot (\mathbf{W}^{\mathbf{Fo}}\mathbf{H}^{\mathbf{Fo}}) + \mathbf{W}^{\mathbf{B}}\mathbf{H}^{\mathbf{B}}$$

will be used to learn the timbre features iteratively in $\mathbf{H}^{\mathbf{\Gamma}}$:

1. Initialize $\Theta_s \equiv \Theta \backslash \mathbf{H}_{\mathbf{v}}^{\mathbf{\Gamma}}$ for each audio sample s. Initialise the global $\mathbf{H}_{\mathbf{v}}^{\mathbf{\Gamma}}$ to be updated for all samples.
2. For each global iteration, go over each audio sample once.
3. For each audio sample s, update Θ_s and $\mathbf{H}_{\mathbf{v}}^{\mathbf{\Gamma}}$ using the NMF update equations.
4. For each global iteration, the order of audio samples should randomised.
5. Iterate till convergence of $\mathbf{H}_{\mathbf{v}}^{\mathbf{\Gamma}}$

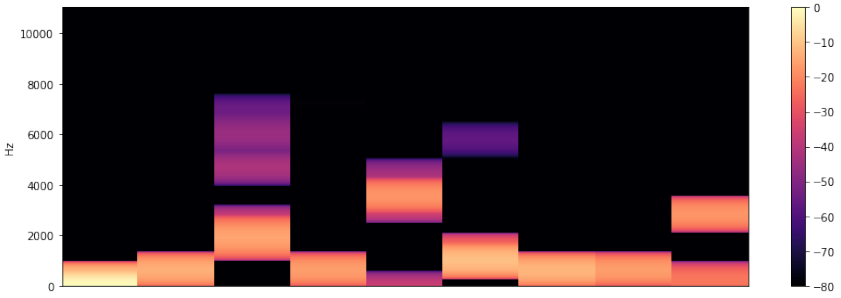
IS divergence between successive $\mathbf{H}^{\mathbf{\Gamma}}$ was checked for convergence.
The final timbre feature matrix for a particular singer/instrument $\mathbf{V}$ will be:

$$\mathbf{W}_{\mathbf{v}}^{\mathbf{\Phi}} = \mathbf{W}^{\mathbf{\Gamma}}\mathbf{H}_{\mathbf{v}}^{\mathbf{\Gamma}}$$

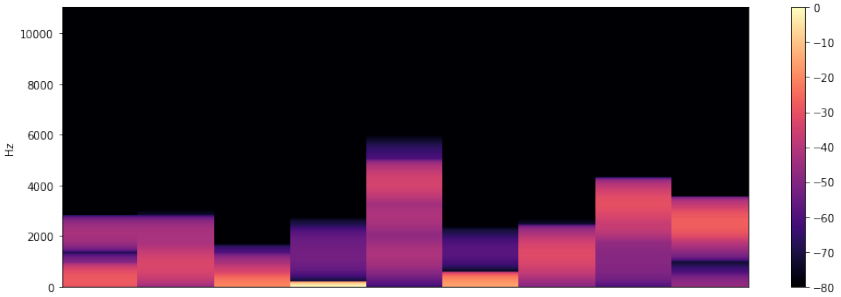
This matrix then used to determine the pitch salience representation of music containing the dominant singer $\mathbf{V}$.

To make the vocal melody extraction more tolerant to instruments, timbre features for background music, in this case instruments, are also extracted using the same algorithm as above. Audio samples containing actual background music and simple polyphonic instrumental music was used to do so. Let us denote background by $\mathbf{B}$. Thus, we will learn the weights $\mathbf{H}_{\mathbf{B}}^{\mathbf{\Gamma}}$.
The final timbre feature matrix for background music will be:

$$\mathbf{W}_{\mathbf{B}}^{\mathbf{\Phi}} = \mathbf{W}^{\mathbf{\Gamma}}\mathbf{H}_{\mathbf{B}}^{\mathbf{\Gamma}}$$



(a) $\mathbf{W_v^\Phi}$ for a singer



(b) $\mathbf{W_B^\Phi}$ for background music

Figure 5: Extracted Timbre features

4.3 Pitch Salience Extraction using Timbre Features

Let us re-write the SF-NMF model using the extracted timbre features:

$$\mathbf{S^X} = \mathbf{S_v^\Phi} \odot \mathbf{S_v^{Fo}} + \mathbf{S_B^\Phi} \odot \mathbf{S_B^{Fo}} + \mathbf{S^R}$$
$$\mathbf{S^X} = (\mathbf{W_v^\Phi H_v^\Phi}) \odot (\mathbf{W^{Fo} H_v^{Fo}}) + (\mathbf{W_B^\Phi H_B^\Phi}) \odot (\mathbf{W^{Fo} H_B^{Fo}}) + \mathbf{W^R H^R}$$

This model estimates the pitch salience representation of the vocal melody in $\mathbf{H_v^{Fo}}$. Here, $\mathbf{R}$ denotes the residual spectrum. Since we are not estimating both vocal ans background timbre features, we need another feature matrix for the residual sounds. not covered by either of the timbre features. Thus, we are not learning the timbre features along with the pitch salience, but instead enforcing the learnt timbre features on the pitch salience.

The new NMF iterative update equations for to estimate the unknown parameters Θ_s are:

$$\Theta_s = [\mathbf{H_v^\Phi}, \mathbf{H_v^{Fo}}, \mathbf{H_B^\Phi}, \mathbf{H_B^{Fo}}, \mathbf{W^R}, \mathbf{H^R}]$$

$$\mathbf{H_v^{F_0}} \leftarrow \mathbf{H_v^{F_0}} \odot \frac{(\mathbf{W^{F_0}})^\top (\mathbf{S_v^\Phi} \odot (\mathbf{S^X})^{\cdot(\beta-2)} \odot \mathbf{S^{X_0}})}{(\mathbf{W^{F_0}})^\top (\mathbf{S_v^\Phi} \odot (\mathbf{S^X})^{\cdot(\beta-1)})}$$

$$\mathbf{H_v^\Phi} \leftarrow \mathbf{H_v^\Phi} \odot \frac{(\mathbf{W_v^\Phi})^\top (\mathbf{S_v^{F_0}} \odot (\mathbf{S^X})^{\cdot(\beta-2)} \odot \mathbf{S^{X_0}})}{(\mathbf{W_v^\Phi})^\top (\mathbf{S_v^{F_0}} \odot (\mathbf{S^X})^{\cdot(\beta-1)})}$$

$$\mathbf{H_B^{F_0}} \leftarrow \mathbf{H_B^{F_0}} \odot \frac{(\mathbf{W^{F_0}})^\top (\mathbf{S_B^\Phi} \odot (\mathbf{S^X})^{\cdot(\beta-2)} \odot \mathbf{S^{X_0}})}{(\mathbf{W^{F_0}})^\top (\mathbf{S_B^\Phi} \odot (\mathbf{S^X})^{\cdot(\beta-1)})}$$

$$\mathbf{H_B^\Phi} \leftarrow \mathbf{H_B^\Phi} \odot \frac{(\mathbf{W_B^\Phi})^\top (\mathbf{S_B^{F_0}} \odot (\mathbf{S^X})^{\cdot(\beta-2)} \odot \mathbf{S^{X_0}})}{(\mathbf{W_B^\Phi})^\top (\mathbf{S_B^{F_0}} \odot (\mathbf{S^X})^{\cdot(\beta-1)})}$$

$$\mathbf{H^R} \leftarrow \mathbf{H^R} \odot \frac{(\mathbf{W^R})^\top ((\mathbf{S^X})^{\cdot(\beta-2)} \odot \mathbf{S^{X_0}})}{(\mathbf{W^R})^\top (\mathbf{S^X})^{\cdot(\beta-1)}}$$

$$\mathbf{W^R} \leftarrow \mathbf{W^R} \odot \frac{((\mathbf{S^X})^{\cdot(\beta-2)} \odot \mathbf{S^{X_0}})(\mathbf{H^R})^\top}{(\mathbf{S^X})^{\cdot(\beta-1)}(\mathbf{H^R})^\top}$$

4.4 Other Advantages of Extended SF-NMF model

From this model:

$$\mathbf{S^X} = \mathbf{S_v^\Phi} \odot \mathbf{S_v^{F_0}} + \mathbf{S_B^\Phi} \odot \mathbf{S_B^{F_0}} + \mathbf{S^R}$$

$$\mathbf{S^X} = (\mathbf{W_v^\Phi H_v^\Phi}) \odot (\mathbf{W^{F_0} H_v^{F_0}}) + (\mathbf{W_B^\Phi H_B^\Phi}) \odot (\mathbf{W^{F_0} H_B^{F_0}}) + \mathbf{W^R H^R}$$

the melody can even be separated from the polyphonic audio, simply by taking the inverse STFT of $\mathbf{S_v^\Phi} \odot \mathbf{S_v^{F_0}}$. This was not possible in SF-NMF because $\mathbf{W^\Phi}$ contained timbre features of the background music as well.

5 Experiments and Results[1]

For vocal melody extraction, the dataset **MIR-1K** was used. It has song clips from different artists, where the singing voice and accompaniment music are recorded in different channels, thus can be used separately.
For instrumental melody extraction, **Bach-10** dataset was used, having 10 instrumental songs with 4 instruments, along with there pitch annotations.

5.1 Model Parameters

All the audio clips were sampled at a rate of 22050 samples per second. For the spectrogram $\mathbf{S^{X_0}}$, a window size of 100 ms was taken to enhance the frequency resolution, and a hop size of 256 samples was chosen.
The timbre features were decomposed as:

$$\mathbf{S^\Phi} = (\mathbf{W^\Gamma H_v^\Gamma H^\Phi})$$

$$s_{fn}^\Phi = \sum_k^K h_{kn}^\Phi \Phi_k(f) = \sum_k^K h_{kn}^\Phi (\sum_p^P h_{pk}^\Gamma \Gamma_p(f))$$

Here, $P = 30$ and $K = 10$.
As for the fundamental frequencies chosen in $\mathbf{S^{F_0}} = \mathbf{W^{F_0} H^{F_0}}$, $\mathbf{H^{F_0}}$ is a $U \times N$ matrix, where N is the number of time bins in the spectrum, and U is the number of fundamental frequencies chosen. I chose $U = 301$ fundamental frequencies between the notes C2 (65 Hz) and C7 (2093 Hz), thus keeping 5 frequencies per semitone (U_{st}).

[1]All the code used for the experiments is available at `https://github.com/gargrohin/Melody-Extraction-from-Polyphonic-Music`

5.2 Evaluation Metrics

The following metrics are used to evaluate the performance of extended SF-NMF model: [7]

1. Raw Pitch Accuracy (RPA) : The proportion of melody frames in the ground truth for which predicted fundamental frequency is correct (within half a semitone of the ground frequency).
2. Raw Chroma Accuracy (RCA) : As raw pitch accuracy, except that both the estimated and ground truth frequency sequences are mapped onto a single octave. This gives a measure of pitch accuracy that ignores octave errors, a common error made by melody extraction systems.

5.3 Vocal Melody

From the **MIR-1K** dataset, I separated the vocals and the background music from 80% of the clips. Using the separated clips, I extracted the timbre features for different singers. An example is shown in figure 4.
Then using the Extended SF-NMF model, the pitch-salience representations were estimated for the clips from the 20% category.
Figure 5 shows $\mathbf{H^{Fo}}$ for a clip from a female singer 'amy'.

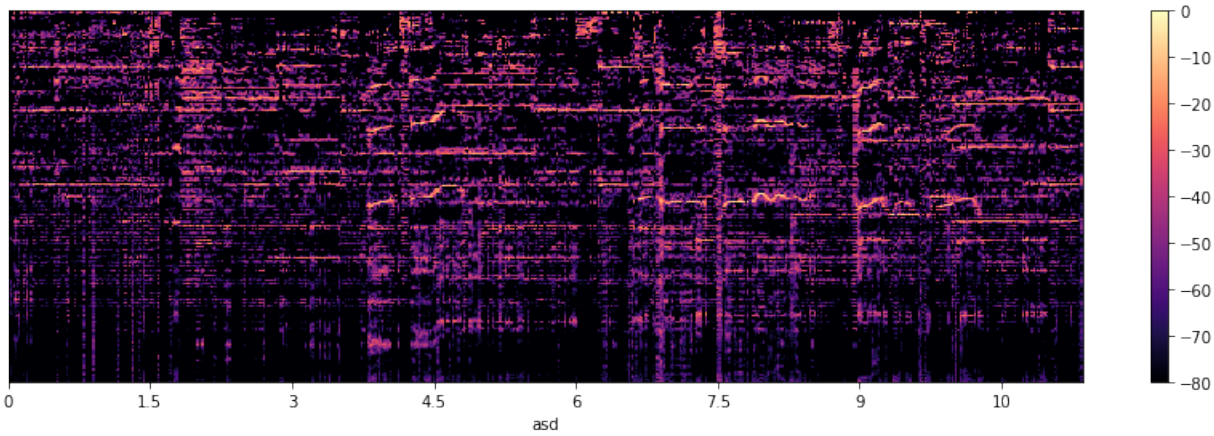


Figure 6: Pitch Salience with pre-trained timbre features

For the same clips, I used the original SF-NMF model to estimate the pitch salience representation. Figure 6 shows that for the same clip as above.

It can be observed that pre-trained timbre features are able to filter out the background sounds, specially in the lower frequency range. In order to do a quantitative analysis of pitch salience representations, I extracted an estimate of melody from them by taking the maximum fundamental frequency with the maximum energy above a threshold for each time bin. (Figure 7)
I used the evaluation metrics mentioned above to compare the performance of the 2 pitch-salience representations.

5.4 Results on Vocal Melody

Raw Pitch Accuracy (RPA) and Raw Chroma Accuracy (RCA) were calculated for both pitch salience representations.

I took the average accuracy scores over the clips, and thus the difference:

Extended SF-NMF gives a significant improvement to the RCA, and on some clips, to RPA as well.

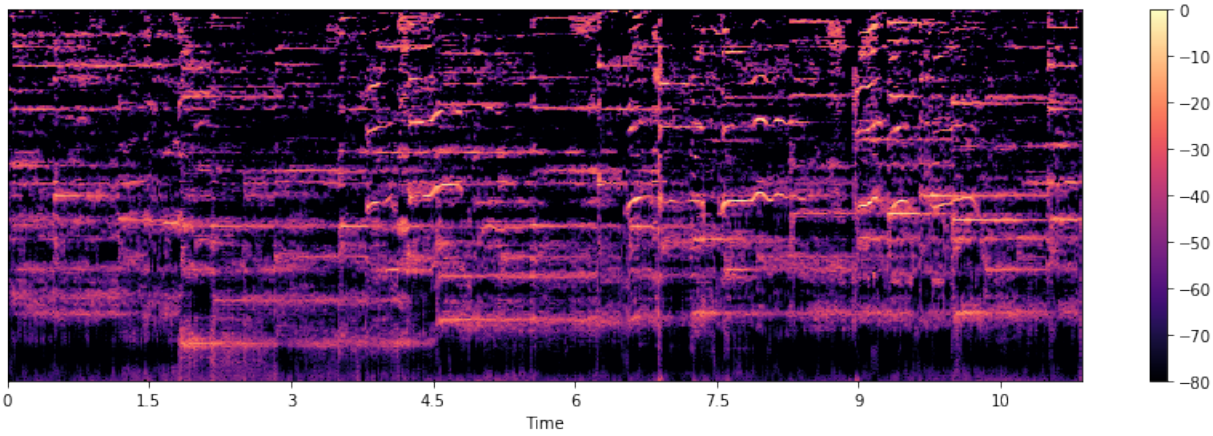


Figure 7: Pitch Salience without pre-trained timbre features

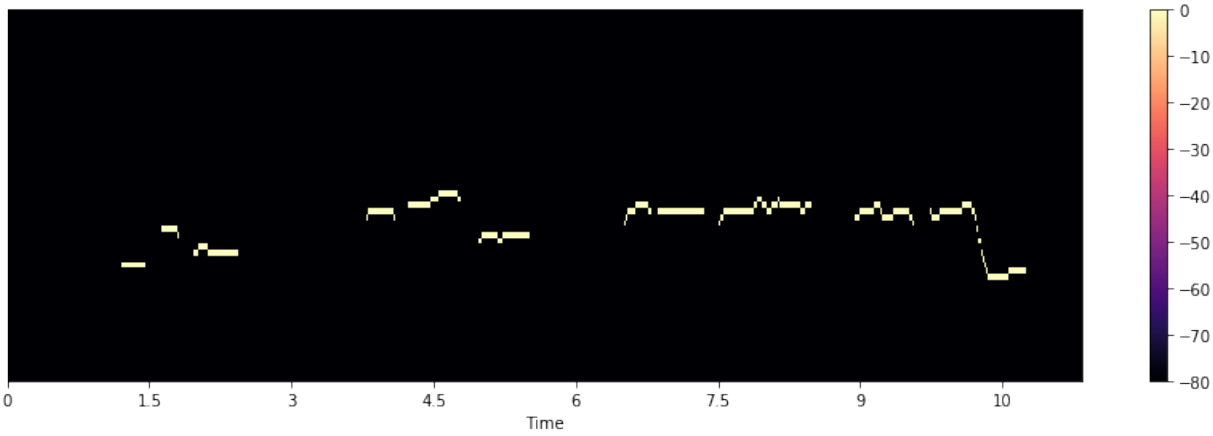


Figure 8: Ground vocal melody for the example clip

Average Scores		
Accuracy Metric	Without timbre features	With timbre features
RCA	0.48	0.55
RPA	0.40	0.42

Table 1: Average Accuracy scores

Scores with highest difference		
Accuracy Metric	Without timbre features	With timbre features
RCA	0.38	0.50
RPA	0.25	0.32

Table 2: Scores with highest difference

5.5 Instrumental Melody

I used the **Bach-10** dataset for instrumental melody extraction. It contains classical instrumental pieces, with 4 instruments: violin, saxophone, bassoon and clarinet, which can also be separated. I focused to extracting 2 stronger melodies present: saxophone and violin. Although the violin melody sounds more dominant to a casual listener, musicians may desire the the extracted melody to be of either of the 2 instruments. This is where the flexibility in controlling the melody line

provided by Melody specific extended SF-NMF model helps.

The melody extracted by SF-NMF model cannot be controlled. We cannot specify the instrument which it has to follow, as it only gives a unique pitch salience representation for an audio clip. Whereas for the Melody specific SF-NMF, I extracted the timbre features for both violin and saxophone,and their respective background features from the dataset. Using them, I was able to extract 2 different pitch salience representations. A quantitative analysis of the representations:

5.6 Results on Instrumental Melody

Scores for Violin melody		
Accuracy Metric	Without timbre features	With timbre features
RCA	0.58	0. 62
RPA	0.47	0.44

Table 3: Scores for Violin melody

Scores for Saxophone melody		
Accuracy Metric	Without timbre features	With timbre features
RCA	0.25	0.40
RPA	0.15	0.29

Table 4: Scores for Saxophone melody

Clearly, the original model only focuses on what seems to be the dominant melody, the violin. It fail for the other instruments. Whereas, the accuracy difference for saxophone shows that if timbre features for the instrument are estimated beforehand and used the way I have, then we can extract the melody line of that instrument, thus controlling the pitch salience to by timbre features. This melody is only partly extracted by the original SF-NMF model, and does not differentiate between the 2 melodies.

6 Conclusions and Future Work

This report addresses the problem of melody extraction from polyphonic music using data driven methods. In this work, the pitch salience representation obtained from SF-NMF model has been modified so that it can be used further for melody extraction. The modifications solve the problem of subjectivity of melody, and gives more freedom in deciding which melody to extract by taking advantage of the timbre features extracted. I have shown that the modified pitch salience representation works slightly better than the baseline for vocal melody.

The major improvement is in the instrumental melody, where the melody line of the instrument which does not have a dominant presence, can also be extracted with decent raw chroma accuracy.

For future works, the second stage of the pipeline, that is constructing fundamental frequency determination from the pitch salience. Currently, neural networks like CRNN described in seection 2 are used. They can be improved by taking inspiration from the recent advancements in deep learning in the field of audio signal processing and music information retrieval.
Jointly training both the extended SF-NMF model with the meural network might result in better overall performance.

7 References

[1] Deep Salience Representations for F0 Estimation in Polyphonic Music.
Rachel M. Bittner and Brian McFee and Justin Salamon and Peter Li and Juan Pablo Bello, ISMIR 2017.

[2] Main Melody Estimation with Source-Filter NMF and CRNN.
Dogac Basaran and Slim Essid and Geoffroy Peeters, ISMIR 2018.

[3] A Musically Motivated Mid-Level Representation for Pitch Estimation and Musical Audio Source Separation.
Jean-Louis Durrieu, Bertrand David, Gal Richard, IEEE Journal of Selected Topics in Signal Processing 2011.

[4] Algorithms for non-negative matrix factorization.
Daniel D. Lee, H. Sebastian Seung, NIPS'00 Proceedings of the 13th International Conference on Neural Information Processing Systems, 2000.

[5] Multimedia Information Retrieval lab, 1000 song clips, dataset for singing voice separation.
Chao-Ling Hsu and Prof. Jyh-Shing Roger Jang

[6] Bach 10 Dataset—- A Versatile Polyphonic Music Dataset.
Zhiyao Duan, Bryan Pardo, 2012

[7] Melody Extraction from Polyphonic Music Signals: Approaches, applications, and challenges.
Justin Salamon ; Emilia Gomez ; Daniel P. W. Ellis ; Gael Richard. IEEE Signal Processing Magazine, 2014