Methods for Polyphonic Music Transcription

Jeremy Nash[a], Paul Schroeder[a], Mark Liu[b]

[a] *Electrical Engineering Department, University of Michigan, Ann Arbor, MI 48109*
[b] *Computer Science Department, University of Michigan, Ann Arbor, MI 48109*

Abstract

Music transcription is the process of converting an audio recording into a musical score. Musicians have been transcribing music manually for centuries to learn new songs and gain musical insight. Transcription, however, is an arduous manual process for musicians, and it's a difficult problem for computers because the pitches in a musical score are the human perceptions of a time-varying and frequency-varying physical phenomenon. There has been a flurry of research on how to solve this problem. We investigate the best solutions to the automated transcription problem and evaluate and compare their performance.

Keywords: Transcription, sparse coding, non-negative matrix factorization, Bayesian non-parametrics, music information retrieval

1. Introduction

1.1. Terminology

Semitones are the smallest music interval in Western music. On a piano, semitones are the adjacent keys.

Timbre represents the spectral envelope, time envelope, noise, tonality, and onset of a musical note. Timbre captures the unique properties of a musical instrument.

Pitches are represented by the notes used in musical notation. Pitches are a psychological perception and not an objective physical property. While we can intuitively assign a linear order to pitches (by referring to pitches as lower or higher than other pitches), pitches are rarely a single frequency and are instead a function of their timbre.

Polyphony means multiple simultaneous notes. Monophony means only a single note at a time. These terms carry slightly different meanings outside of automated transcription research.

2. Motivation

Applications of a music transcription are versatile. Transcription produces a compact and standardized parametric representation of music. Such representation is needed for content-based retrieval of music in most current musical databases. It is useful in music analysis systems for tasks such as melody extraction, music segmentation and rhythm tracking. Transcription aids musicologists in analyzing music that has never been written down, such as improvised or ethnical music. The conversion of an acoustical waveform into parametric description is also useful in the process of making music, as well as in newer coding standards, such as MPEG-4, which may include such descriptions.

3. Problem Statement

In Western music, each note corresponds to a single, fundamental frequency. The musical note A above middle C corresponds to 440Hz. If this were the complete story, there would be a one-to-one correspondence between frequencies and notes, and the transcription problem would be as simple as converting the frequencies into notes. The reality, however, is that each instrument has a unique timbre with a set of overtones that vary over time and overlap with the frequencies of other notes.

Like most research in automatic music transcription, we restrict our problem to transcribing polyphonic piano music. The reason for this is primarily the availability of online piano MIDI databases and the popularity of the piano in Western music. At the same time, the piano has a complex timbre and adequately represents the difficulties in the generic transcription problem.

3.1. Piano physics

?] details the physics of piano sound synthesis. We will just note here that...

Figure X shows the spectral profile of a low C note played over a couple seconds.

This is typically the lowest (But it's more complicated than that...) - Show why it's complicated - Show some of the physics

Email addresses: nashj@umich.edu (Jeremy Nash), pschro@umich.edu (Paul Schroeder), markmliu@umich.edu (Mark Liu)

4. Related Work

5. Methodologies

There are currently two main approaches to polyphonic music transcription. The first approach learns a supervised learning model that uses frequency magnitudes to predict the presence of a note. The second approach uses unsupervised learning techniques to extract

5.1. Audio data

Our training data comes from a large database of piano MIDI files. MIDI is a standard for specifying...

5.2. Spectrogram

In all of the methods we explored, we computed a spectrogram of the wav file with a window size of 128ms, a Hanning windowing function, and a 10ms hop between frames. This window size gives us a frequency resolution of 8Hz, which is slightly inadequate for resolving the fundamental frequencies of lower notes and overly resolute for the higher frequencies due to the logarithmic spacing of the semitone fundamental frequencies.

To the authors' knowledge, all of the methods in the transcription literature compute a spectrogram as a preprocessing step before applying their specialized algorithm. In the NMF methods, the spectrogram matrix is decomposed into a piano roll matrix and the note spectrum dictionary matrix. In the supervised classifier methods, the classifiers use features from the spectrogram matrix to predict the presence of a note.

5.3. Constant Q transform

Unlike the linear frequency scale used in an FFT, semitones in Western music are spaced logarithmically. This logarithmic spacing derives from our roughly logarithmic perception of pitches, first captured by Stevens et al. [1] through the mel scale.

Transforming the spectra to a logarithmic frequency scale achieves two main benefits. It reduces the dimensionality of the training data, since most of the spectral energy occupies a logarithmic grid of frequencies. In practice, Bock and Schedl [2] were able to reduce the dimensionality of their training data from 2048 FFT magnitudes down to 183.

The logarithmic frequency scale also decreases sensitivty to detuning. In a linear frequency scale, detuning, especially in higher semitones, will cause spectral energy to leak into neighboring frequency bins, creating more variability in the training data. On a logarithmic frequency scale, minor detuning would result in less spectral energy leakage and less unnecessary variability in the training data.

5.4. Note onset detection

This method proposes restricting the problem to determining note onsets. This is a simpler problem than detecting note durations because of the possibility for time-varying timbres in instruments. Most researchers in this field evaluate their music transcription methods on their note onset detection performance.

5.5. Discriminative models

5.6. SVM

Poliner and Ellis [3]

5.7. Restricted Boltzmann machine

Nam et al. [4] proposes ... training a restricted Boltzmann machine on the spectrogram features. The features computed by the RBM are used by an SVM

5.8. Boosted classifier

?] proposes a cascade of boosted classifiers.

5.9. Feed forward neural network

5.10. Non-negative matrix factorization

Non-negative matrix factorization (NMF) is an unsupervised, dimensionality reduction technique that factors a non-negative matrix V with dimensions $F \times N$ into two non-negative matrices W and H.

$$V \approx WH$$

where W is an $F \times K$ matrix and H is a $K \times N$ matrix. K is typically chosen so that $F * K + K * N << F * N$.

Typically this factorization is performmed with the goal of recovering some useful structure or information from the matrix factors. NMF has been applied to polyphonic transcription and other applications [5]. NMF also has been used for feature extraction for use with other general-purpose classifiers (e.g. SVMs) for transcription.

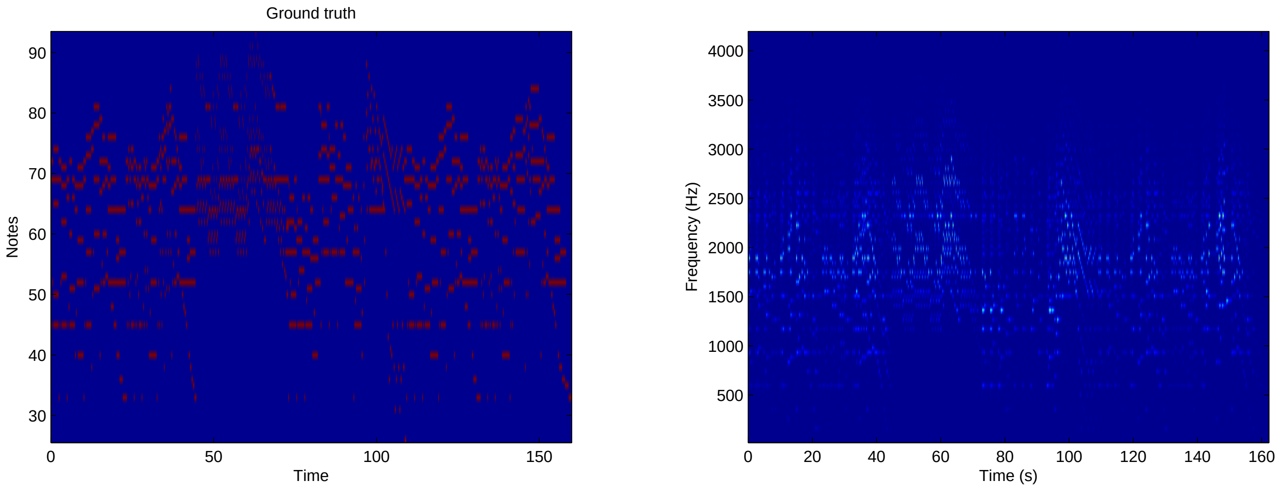
NMF is usually posed as a minimization problem. There are two cost functions commonly used for NMF. The first is a euclidean distance cost function, defined as the squared difference between the corresponding elements in each matrix:

$$||V - WH||^2 = \sum_{ij} (V_{ij} - (WH)_{ij})^2$$

Another commonly used measure is the KL divergence:

$$D(V||WH) = \sum_{ij} (V_{ij} \log \frac{V_{ij}}{(WH)_{ij}} - V_{ij} + (WH)_{ij})$$

The functions $||V - WH||^2$ and $D(V||WH)$ are convex in W only and H only, but they are not convex in W and H together. There is no known procedure for finding the global minimum of these functions, but Seung and Lee [6] have proposed methods for updating W and H such that



(a)

(b)

Figure 1: (a) shows the piano roll generated from a MIDI file. (b) shows a spectrogram of the wav file using the constant Q transform with a step size of 10ms. One can immediately see the similarity in the shape of the two plots, although the transcription process is complicated by the overlapping frequencies, changing volume, and noise in the spectrogram plot.

the cost functions above will decrease or remain constant after every update. Their paper proposes a multiplicative update rule and a gradient descent rule.

The multiplicative update rule for improving the Euclidean distance error measure $||V - WH||^2$ is:

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu}\frac{(W^TV)_{\alpha\mu}}{(W^TWH)_{\alpha\mu}}, W_{\gamma\alpha} \leftarrow W_{\gamma\alpha}\frac{(VH^T)_{\gamma\alpha}}{(WHH^T)_{\gamma\alpha}}$$

Typically, W and H are initialized with random, nonnegative values and iterated until convergence.

5.11. NMF for Piano Transcription

In piano transcription, V might be a spectrogram with dimensions $2048 \times N$, with N corresponding to the length of the song. W might be 2048×88 and encode the spectral envelope for each piano key in each column of the matrix. Because of this structure, W is sometimes referred to as a note dictionary matrix. H might be $88 \times N$ sparse piano roll matrix. It is this strong interpretability of the matrix factors W,H that makes NMF an attractive technique for transcription. Since H contains the desired note information for transcription, accurately recovering H is the effective goal of NMF for piano transcription. After H is recovered, post-processing such as smoothing and thresholding is usually applied to remove spurious activations and improve overall results.

There are a few problems applying NMF as it is presented above to the piano transcription problem:

- The algorithms from Seung and Lee [6] require that the number of keys present in the song is specified in advance. While a full piano keyboard contains a constant 88 keys, a piano song will often only use a subset of all the keys. In practice, pre-selecting the W matrix to find more keys than are present results in the algorithm discovering parts of notes or the same note at different points in the note's Attack-Decay-Sustain-Release (ADSR) envelope; for example, the attack of a note can have a different spectral signature than its release. Weninger et al. [7] use this fact explicitly to learn notes at different points in the ADSR envelope using NMF as a preprocessing step to SVM classification.] uses cross-validation to select the appropriate number of keys.
- There is no guarentee that a note will have a representation in W, even if it is present in the training data.
- The piano roll matrix H and the spectrogram V are large and often dense matrices, so convergence using the multiplicative update rules can take on the order of minutes on a modern laptop.
- The notes represented in the W and H matrix are not ordered properly. Ordering does not affect raw performance but greatly detracts from the interpretability. One can attempt to estimate the pitch of the columns of M and then sort. Abdallah and Plumbley [8] found that structuring the initial W would lead to a correctly ordered output. Boulanger-Lewandowski et al. [9] resorted to ordering the notes in W by inspection.

5.12. Sparse non-negative matrix factorization

One of the great side-effects of NMF is that it often produces a sparse, parts-based representation. This is desirable in music transcription because the piano roll matrix, H, is a sparse matrix of note activations. However, the NMF method does not contain an explicit sparsity objective, and so the update rules do not always converge to a sparse solution. Hoyer [10] extended NMF to contain a tunable sparsity objective. This sparse NMF method in Hoyer [10] was adopted by Abdallah and Plumbley [8] in their transcription system.

5.13. Weakly Supervised NMF

In the previous NMF methods, we extracted both the note dictionary matrix W and the piano roll from the spectrogram. In piano transcription, however, the dictionary matrix should be similar across multiple transcriptions. We can guide the NMF process by explicity creating a dictionary matrix W off-line. Not only does this alleviate the problem where notes aren't represented in W, but it does not require any manual or automatic sorting that is normally required when W is produced by NMF. Weninger et al. [7] suggest that the W matrix may be constructed by performing NMF with rank one (i.e. K = 1) on recordings of isolated notes, once for each note. The first column of each W matrix is concatenated to form a dictionary matrix for later transcription. One can also construct a dictionary matrix by directly concatenating normalized spectral features[11].

Some have suggested adding additional bases to W. For example, one could add multiple examples of the same note played on different pianos to prevent from overemphasizing the characterisics of a specific instrument [cite FINDTHIS]. Weninger et al. [7] found including different spectral features for the onset, sustain and decay of each note to be useful. In these cases, note activations from different bases but which correspond to the same pitch are summed to form $88 \times N$ piano roll matrix.

Once W is determined, the next step is to recover the piano roll matrix H. W can be used as an initial value in a standard NMF formulation. Another way to recover H is to use the standard multiplicative update rule for H and skip the updating of W since it is already known. Niedermayer [11] refers to this variant of NMF as Non-Negative Matrix *Division*, as we're only estimating H in the approximation.

5.14. Our Implementation

We implemented a NMF transcription system. We used constructed our W matrix by concatenating spectra from training recording. We tried both a standard STFT spectrogram and constant Q transform spectrogram. We use the normalizations proposed by Niedermayer [11]. After creating the W matrix, we calculate the H matrix using the previously-defined multiplicative update rule. The H matrix is smoothed using a median filter and thresholded. [TODO: Results]

5.15. Bayesian nonparametric models

One of the downsides of the NMF methods is that they require the number of notes (or sources) to be specified

5.16. Smoothing

The raw output of many transcription techniques is often noisey and contains many short, spurious notes. Additionally, otherwise correctly transcribed notes may contain small unwanted gaps. Obviously these detracts from the quality of the transcription. Smoothing can be used to improve the transcription by removing the small errors described above. Some smoothing techniques are as simple as moving average or median filters, others take advantage of prior knowledge of the temporal structure of music.

5.16.1. HMM smoothing

In many of the techniques used (such as SVM's), notes at a single time step are classified independently of the notes at adjacent time steps. Since these methods do not take advantage of the temporal structure of piano music, hidden Markov models are often used for post-processing the output of the discriminative models. We model each note independently using a two-state HMM, where the states correspond to the note being on or off. The transition matrix and state priors can be estimated from the ground truth piano roll, while the emission matrix can be estimated using the output from the classifier as the observable variables. Applying the Viterbi algorithm to each note class results in a smoother piano roll that eliminates unlikely, spurious, isolated notes.

5.16.2. Probabilistic spectral smoothness

5.17. Recurrent neural network

6. Evaluation

6.1. MIDI databases

The MIDI music used to evaluate these methods came from the Classical Piano MIDI Page at http://www.piano-midi.de/, a free, online database of piano MIDI files with a wide range of composers. This database, along with the MAPS database, are a popular choice for piano transcription researchers. Table X gives a list of the composers and songs we used for testing.

The MIDI files represent the ground truth data. To generate training data, we synthesized wav files from the MIDI files using Timidity...

Synthesized with Timidity -¿ 16kHz

6.2. Evaluation measures

There is no standard metric to evaluate the performance of a transcription method. However, many researchers use the following methods:

6.2.1. Overall accuracy

$$Acc = \frac{TP}{TP + FP + FN}$$

where TP (true positives) is the number of correctly transcribed notes (over all notes), FP (false positives) is the number of unvoiced frames transcribed as voiced, and FN (false negatives) is the number of voiced frames transcribed as unvoiced. Note that notes that are missed and notes that are inserted are weighed equally.

6.3. Frame-level transcription error score

In the previous measure "double counts" notes transcribed correctly but at the wrong time. Frame-level transcription error score seeks to avoid that. At each time frame, let N_{sys} be the number of reported pitches, N_{ref} the ground truth number of pitches, and N_{corr}, their intersection, is the number of correctly reported pitches. Then we define

$$E_{tot} = \frac{\sum_{t=1}^{T} \max(N_{ref}(t), N_{sys}(t)) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)}$$

E_{tot} can be further split into three separate scores: E_{sub}, which counts the number (at each frame) of ground truth notes for which some other note was reported, E_{miss}, the number of ground truth notes which cannot be accounted for, and E_{fa}, the number of reported notes which cannot be paired with a ground truth note. The equations for these, respectively, are

$$E_{sub} = \frac{\sum_{t=1}^{T} \min(N_{ref}(t), N_{sys}(t)) - N_{corr}(t)}{\sum_{t=1}^{T} N_{ref}(t)}$$

$$E_{miss} = \frac{\sum_{t=1}^{T} \max(0, N_{ref}(t)) - N_{sys}(t)}{\sum_{t=1}^{T} N_{ref}(t)}$$

$$E_{fa} = \frac{\sum_{t=1}^{T} \max(0, N_{sys}(t)) - N_{ref}(t)}{\sum_{t=1}^{T} N_{ref}(t)}$$

6.3.1. Note Onset Detection

Frame-level transcription error requires a high level of precision. An error-measure perhaps more fitting for this problem is note onset detection, since onset is usually more important than offset. To be counted as correct, our transcribed note must be turned "on" within 100 milliseconds of the ground truth onset.

6.4. Method comparison

6.4.1. Reported measures

Algorithm	Accuracy	Score
SVM	21%	21
SVM with HMM	22%	22
NMF	23%	23
Sparse NMF	24%	24
AdaBoost	25%	25

6.4.2. Our evaluation

The following table captures our evaluation of these methods. We expect these accuracy measures to be lower than they were reported in the original papers because we are using a smaller training dataset and not attempting to optimize the methods.

Algorithm	Accuracy	Score
SVM	18%	18
SVM with HMM	19%	19
NMF	20%	20
Sparse NMF	21%	21
AdaBoost	22%	22

7. Conclusion

8. Individual Effort

- **Jeremy** wrote and evaluted the SVM method in Poliner and Ellis [3], the Bayesian nonparametric method in Blei et al. [12], and the sparse non-negative matrix factorization method in Abdallah and Plumbley [8]. Jeremy also wrote the paper.
- **Mark** implemented the hidden Markov model in Poliner and Ellis [3] and the LSTM network in Bock and Schedl [2].
- **Paul** implemented the LSTM network in Bock and Schedl [2].

References

[1] S. S. Stevens, J. Volkmann, E. B. Newman, A scale for the measurement of the psychological magnitude pitch, The Journal of the Acoustical Society of America 8 (1937) 185.

[2] S. Bock, M. Schedl, Polyphonic piano note transcription with recurrent neural networks, in: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, pp. 121–124.

[3] G. E. Poliner, D. P. Ellis, A discriminative model for polyphonic piano transcription, EURASIP Journal on Advances in Signal Processing 2007 (2006).

[4] J. Nam, J. Ngiam, H. Lee, M. Slaney, A classification-based polyphonic piano transcription approach using learned feature representations., in: ISMIR, pp. 175–180.

[5] P. Smaragdis, J. C. Brown, Non-negative matrix factorization for polyphonic music transcription, in: Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on., IEEE, pp. 177–180.

[6] D. Seung, L. Lee, Algorithms for non-negative matrix factorization, Advances in neural information processing systems 13 (2001) 556–562.

[7] F. Weninger, C. Kirst, B. Schuller, H.-J. Bungartz, A discriminative approach to polyphonic piano note transcription using supervised non-negative matrix factorization, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, pp. 6–10.

[8] S. A. Abdallah, M. D. Plumbley, Polyphonic music transcription by non-negative sparse coding of power spectra, in: Proc. 5th Intl Conf. on Music Information Retrieval (ISMIR), pp. 10–14.

[9] N. Boulanger-Lewandowski, Y. Bengio, P. Vincent, Discriminative non-negative matrix factorization for multiple pitch estimation, in: Proceedings of the 13th International Society for Music Information Retrieval Conference, Porto, Portugal.

[10] P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, The Journal of Machine Learning Research 5 (2004) 1457–1469.

[11] B. Niedermayer, Non-negative matrix division for the automatic transcription of polyphonic music., in: J. P. Bello, E. Chew, D. Turnbull (Eds.), ISMIR, pp. 544–549.

[12] D. M. Blei, P. R. Cook, M. Hoffman, Bayesian nonparametric matrix factorization for recorded music, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), pp. 439–446.