Polyphonic Music Transcription
A Deep Learning Approach

Aniruddha Zalani & Ayush Mittal

April 24, 2014

Abstract

In polyphonic music, many notes are played at once. Transcribing notes from the polyphonic music can help in plagirism detection, artist identification, Genre Classification, Composition Assitance and Music Tutoring Systems. Since, many notes are played at once, therefore, the techniques of multi class classification are not applicable here. In this project, we have learned 88 binary classifier which helps in transcribing notes of polyphonic music. Each classifier detects the presence of one note in the music at every time step. Unsupervised feature learning using RNN-RBM (Recursive Neural Networks and Restricted Boltzmann Machine) and Covolutional Deep Belief Networks has been done. SVM classifiers are build using one-vs-all classification. HMM smoothing has been done to improve the results.

1 Introduction

Polyphonic music in piano means that there are two or more independent notes playing on the same time, in contrast to the monophonic music where only one node is played at a time. A lot work has been done on monophonic transcription but the problem of Polyphonic transcription is still open. Many naturally occuring phenomeno have complex sequences that are inherently sequential but the value at next time step cannot be determined only by the knowledge of previous time step. Examples of such phenomeno inculde music, speech, human motion. Most of them are spanned over high dimensional spaces. Word notes appear together in correlated patterns so this aects the conditional probablity. Many notes are played at once, therefore, techniques of multi class classification are not applicable. Some interesting work on this problem has been done using non-negative matrix factorization method[1][**?**]. Most of the recent work involved use of deep learning methods for unsupervised feature learning. Our approach is based on works of Nicholas et al.,[3] for feature learning. For classification we have used Poliner and Ellis[**?**] SVM based one-vs-all classification. Taking inspiration from music information retrieval using Convolutional Deep Belief Network(CDBN)[**?**], we have proposed CDBN based approach for polyphonic music transcription.

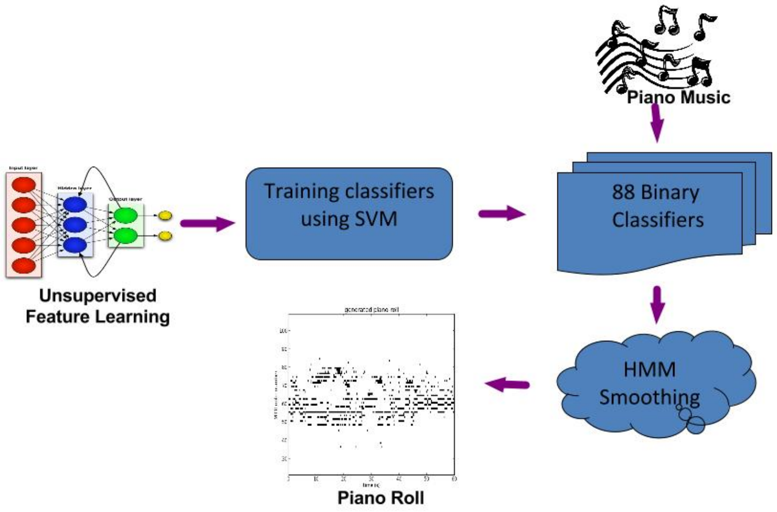

Figure 1: Our Methodology

2 Datasets

We have used a subset of MAPS dataset. Training data comprised of 6 piano files, with nearly thirty minutes of music. Test data comprised of 4 files with nearly 15 minutes of music. We have used a separate cross validation for each SVM classifier.

3 Methodology

3.1 Feature Learning

We use Recursive Neural Networks(RNN) stacked with Restricted Boltzmann Machines(RBM) for unsupervised feature learning.

3.1.1 Restricted Boltzmann Machine

Restricted Boltzmann Machine [**?**] is an energy based model used for learning probability distributions over the input data. An RBM consists of two layers, the input layer and the hidden layer as shown in the figure 9. Energy function for RBM is linear in terms of its free parameters.

3.1.2 Recursive Neural Networks

Recursive Neural Networks(RNN) are the deep nets in which the connections between units forms the directed cycles as shown in the 3. This directed cycles

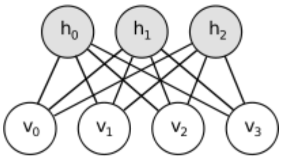

Figure 2: Restricted Boltzmann Machine.
v_i represents visible units and h_i represents hidden units.
src:`www.deeplearning.net/tutorial/rbm.html`

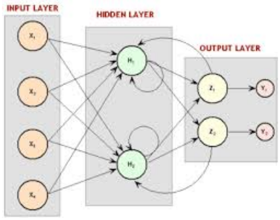

Figure 3: Recursive Neural Network.
src: `http://www.information-management.com`

help in modelling temoral dependecies and hence it is very usefull in our task as the temporal dependencies between different notes is very high in piano music.

3.1.3 RNN-RBM

We stack RBMs to form a Recursive Neural Network. It is an energy based model used for density estimation for temporal sequences. In an RNN, the connections between units forms a directed cycles which helps in modelling tempraol dependencies through our neural network. Figure 4 shows RNN RBM model unrolled over time space, where $v^{(t)}$ is feature vector at time step t, u^t are the hidden units of RNN, ${b_v}^{(t)}, {b_h}^{(t)}$ are the parameters of the RBMs. The probaility distribution is given by:

$$P(\{v^{(t)}\}) = \sum_{t=1}^{T} P(v^{(t)}|\mathcal{A}^{(t)})$$

where $\mathcal{A}^{(t)}$ denotes the sequence history at time t. For training RNN-RBM we have used the implementation given on `www.deeplearning.net`. The implementation uses Stochastic Gradient Descent on every time step for updating parameters. Contrastive Divergence is used in Stochastic Gradient Descent for Gibbs sampling. Instead of random sampling Contrastive Divergence uses the distribution of training data for sampling which leads to faster convergence. We trained our model on 200 epochs for learning the features.

3.1.4 Short Term Fourier Transformation

Besides the features learned from the RNN-RBM model we have also used Short Term Fourier Transformation (STFT) as features for training classifier. Since, STFT determines changes in frequency and phase of local sections of music over different time steps therefore, it is an important feature for diffrentiating different notes. We calculate STFT features of music from the wav files. The

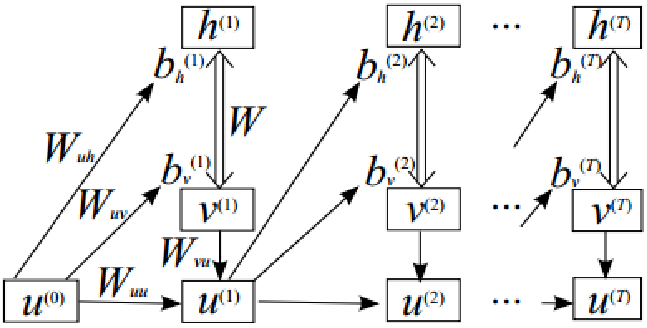

Figure 4: RNN-RBM unrolled over time [3]

wav files and the corresponding midi files are already aligned in the dataset that we are using and therefore, we can associate the STFT features with the features learnt from unsupervised learning.

3.2 Classification

We did a single note training. We used 88 independent support vector machines (SVM). As there were 88 notes into consideration therefore each of the support vector machines corresponded to each note in piano. Supervised training was done for each note individually. We used linear kernel for classification because our data is too large therefore higher order kenrel were more time consuming and only provided insignificant improvement in the results. The results from the RNN-RBM were directly fed into the SVMs and then training was done. Our classification technique is **one-vs-all**.

3.3 Smoothing

The results after svm training can be seen in [figure:7]. Here we can observe that there is a lot of noise so smoothing is required. Each note was modelled independently HMM containing two states. We used forward backward algorithm for smoothing. Here the svm output were seen as posteriors and priors were calculated. HMM smoothing considers the input to svm as a set of observation ($\mathbf{x}$) and hidden variables were the occurance of a particular note(y_i). Therefore the posteriori is $\mathbf{P}(y_i = 1|\ \mathbf{x})$. HMM smoothing is important because zero probabilities can be problamatic as we have a note playing and there is a very tiny gap between them in the graph (of around 10 ms) then chances are that the note nust have been palying continuously and the output of svm is faulty. So, by hmm smothin we fill in that space. Further, by comparing the ground truth and the svm output we can observe that a lot of noise is also present (only single dots). Considering the way that piano is being played, every note that has been played will persist for more than 50-100 ms therefore the single points also correspond to the noise and hence have to be removed via hmm smoothing.

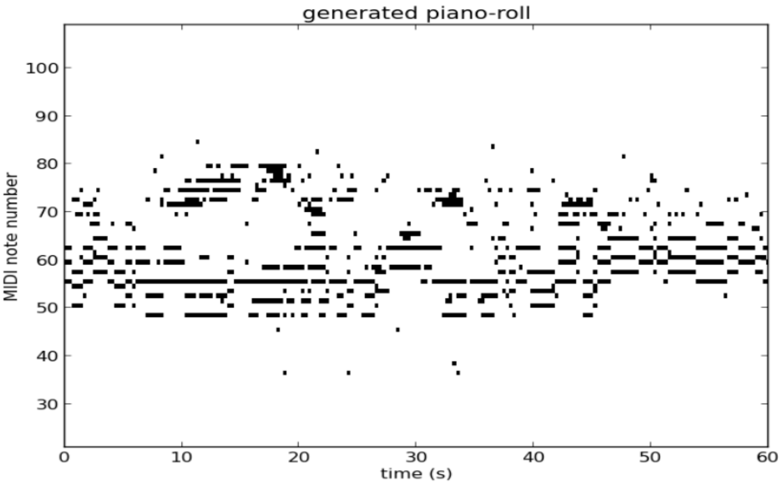

Figure 5: Feature learning.

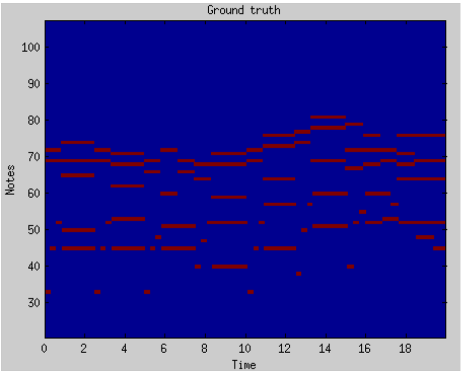

Figure 6: Ground truth

4 Experiments and Results

4.1 Evaluation Metric

- Frame Level Accuracy - TP/(TP + FN + FP)
- Frame-level transcription error score (R_{total})
- E_{sub} number (at each frame) of ground truth notes for which some other note was reported,
- E_{miss} number of ground truth notes which cannot be accounted for.
- E_{fa} the number of reported notes which cannot be paired with a ground truth note.

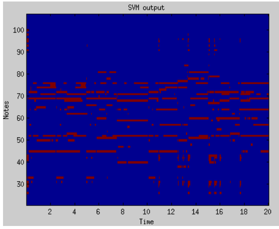

Figure 7: Output of svm

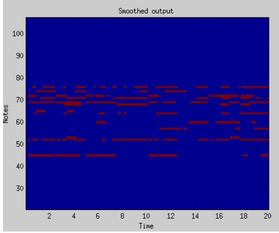

Figure 8: Results aftter HMM postprocessing.

Data	**Accuracy**	E_{tot}	E_{sub}	E_{miss}	E_{fa}
Smooth	0.6310	0.5426	0.1838	0.1570	0.2018
Raw	0.5207	0.8337	0.0951	0.0105	0.7281

Table 1: Results for piano

Algorithm	**Accuracy**
Polinear and Ellis	0.6770
RNN-RBM (Our Approach)	0.6310
Marolt [6]	0.396
Ryyananen and Klapuri [5]	0.4630

Table 2: Comparison from other techniques

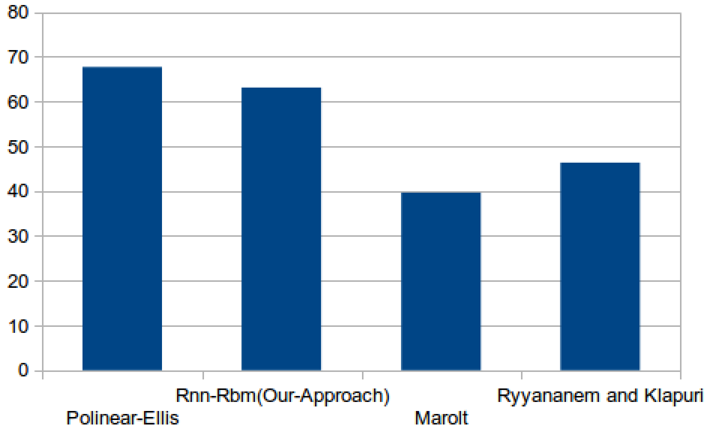

Figure 9: Comparison of accuracies.

4.2 Experiments with Tabla

We collected the Tabla dataset from different websites and we have learnt features from it using RNN-RBM but because of alignment problems in correspoding wav and midi files intermediate tabla roll was poor.

5 Conclusions

We have presented an unsupervised feature learning based approach with SVM classification and HMM smoothing for polyphonic music transcription. Our RNN-RBM based model achieves the accuracies close to state of art techniques. We have achieved 63.1 percent accuracy on piano dataset. Unsupervised feature learning improves results over simple Poliner-Ellis model. We have also experimented with tabla dataset. We also tried to learn features using convolutional deep belief network.

6 Future Work

The feature learning step can be improved through efficient implementations of Convolutional Deep Belief Networks(CDBN). The work can be further extended to various other music devices such as tabla and drums. Also the classification step can be made more efficient by using multi-note training instead of single note training.

7 References

[1] Arnaud , Arshia et al. Real-Time Detection of Overlapping Sound Events with Non-Negative Matrix Factorization

[2] Paris and Judith Non-Negative Matrix Factorization for Polyphonic Music Transcription, IEEE 2003

[3] N. Boulanger-Lewandowski, Y. Bengio and P.Vincent, Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription," ICML, 2012.

[4] J. Nam, J. Ngiam and H. Lee,Classication- Based Polyphonic Piano Transcription Approach Using Learned Feature Representations," ISMIR , pp. 175-180, 2011

[5] M. Ryynanen and A. Klapuri: "Polyphonic music transcription using note event modeling," Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.

[6] M. Marolt: "A connectionist approach to automatic transcription of polyphonic piano music," IEEE Transactions on Multimedia, vol.6, no.3, pp.439–449, 2004.