The Power of Less: Exemplar-based Automatic Transcription of Polyphonic Piano Music

İsmail Arı, Ali Taylan Cemgil*, and Lale Akarun

Bogazici University, Dept. of Computer Engineering, 34342 Istanbul, Turkey
{ismailar,taylan.cemgil,akarun}@boun.edu.tr

Abstract. Transcription of polyphonic piano music is an important computer music problem and many sophisticated methods have been proposed for its solution. However, most techniques cannot fully utilize all the available training data efficiently and do not scale well beyond a certain size. We develop an exemplar-based approach that can easily handle very large training corpora. We maintain transcription performance by only retaining 1% of the training data. The method is competitive with the state-of-the-art techniques in the literature. Besides, it is very efficient and can work in real time.

Keywords: automatic polyphonic music transcription, non-negative matrix factorization, clustering, k-medoids, affinity propagation

1 Introduction

Transcription of music is the process of determining the notes and the time intervals in which they are active given a piece of music. Conventionally, it is done by hand and requires both an educated ear and considerable amount of time. Automation of this process is one of the fundamental problems studied in the field of audio processing [1]. The computational methods developed to solve this problem find application in various areas such as phonetics, speech processing and music information retrieval [1]. Transcription is closely related to pitch detection and tracking which is extensively studied in the literature. Among the proposed methods, the model-based ones on matrix factorization have become very popular in the last decade [1, 2].

It is still not very clear how humans, especially musicians, recognize notes in polyphonic textures. Experience suggests that human listeners become more successful with training in recognizing musical constructs. Inspired partially by this idea, Smaragdis has demonstrated that it is possible to perform polyphonic pitch tracking successfully via a linear model that tries to approximate the observed musical data as a superposition of previously recorded monophonic musical data [3]: $\mathbf{X} \approx \mathbf{DW}$ where $\mathbf{X}$ is the observed spectrogram, $\mathbf{D}$ is the dictionary matrix obtained from the training data, and $\mathbf{W}$ contains the corresponding weights.

* A.T.C. is supported by TÜBİTAK grant 110E292.

Intuitively, we would expect better transcription accuracy with larger and more comprehensive dictionary matrix $\mathbf{D}$, but on the other hand, this potentially very large dictionary leads to an impractical algorithm. A wise solution is to express the dictionary matrix using fewer dimensions in order to perform a faster transcription with a smaller computation and memory requirement. We employ two state-of-the-art exemplar selection algorithms, k-medoids [4] and affinity propagation [5], in order to remove repetitive and irrelevant part of the data. In particular, we significantly reduce the size of the dictionary without compromising the success rate of the full solution.

2 Polyphonic Music Transcription

Let $\mathbf{D}_i$, with elements $D_i(f, \tau_i)$, denote the magnitude spectrogram of monophonic piano recordings belonging to 88 different notes. Here, $i = 1, \ldots, 88$ is the note index, $f = 1, \ldots, F$ is the frequency index, and $\tau_i = 1, \ldots, N_i$ is the time index where F is the number of frequency bins and N_i is the number of columns in $\mathbf{D}_i$. We obtain the training data by concatenating all training vectors, $\mathbf{D} = [\ \mathbf{D}_1\ \mathbf{D}_2\ \ldots\ \mathbf{D}_{88}\ ]$. Test data are composed of polyphonic piano recordings. Let $\mathbf{X}$, with values $X(f, t)$, be the spectrogram of the test data where $t = 1, \ldots, T$ and T is the number of time frames.

We use a basic linear model where the observed spectrum is expressed as a superposition of the training vectors:

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{D}\mathbf{W} \tag{1}$$

The aim is to find the weight matrix $\mathbf{W}$ which minimizes $\mathcal{D}\left[\mathbf{X} \| \mathbf{D}\mathbf{W}\right]$ where $\mathcal{D}[\cdot\|\cdot]$ is a properly selected cost function. We choose KL divergence for the cost function. Note that the model is identical to the NMF model whose update rule is well known [2,6]. We start with random initialization of $\mathbf{W}$, and continue with the following step until convergence:

$$\mathbf{W} \leftarrow \mathbf{W} \odot \left(\frac{\mathbf{D}^\intercal \frac{\mathbf{X}}{(\mathbf{D}\mathbf{W})}}{\mathbf{D}^\intercal \mathbb{1}} \right) \tag{2}$$

The $\odot$ symbol implies element-wise multiplication and division is also done element-wise. $\mathbb{1}$ is a matrix of all ones of the same size with $\mathbf{X}$. Active notes are observed to have significantly higher weights than the inactive ones and they are selected by thresholding. Additionally, each weight value is smoothed by applying a median filter to remove sharp jumps.

Let us now shift our attention to select exemplar data points which are good candidates to represent remaining data points. The classical approach to this problem is the k-medoids method where a medoid of a cluster is defined to be the one whose average dissimilarity to all the objects in the cluster is minimal [4]. In a typical k-medoids algorithm, we start with random initialization of medoids. Then, at each iteration, we assign clusters to the data points and recompute a medoid per cluster until convergence or a permitted maximum

number of iterations. The distance among two data points is usually chosen to be l_1 distance, but any appropriate distance is possible. k-medoids is run several times each with different initial centers and the best solution is taken in order to avoid local optima.

Another approach to clustering via exemplar selection is the affinity propagation (AP) of Frey and Dueck [5]. AP finds a solution by exchanging real-valued messages between data points until a high-quality set of exemplars and corresponding clusters gradually emerges [5]. The number of clusters is not given a priori in AP clustering. It is determined by the preferences: If the preference for selecting a column is larger than its similarity to the nearest possible exemplar, then it is selected as a new exemplar. Similar to k-medoids, AP can use any appropriate similarity measure.

Applying exemplar-based algorithms on the merged data is not practical since it requires pairwise distances of all the columns, which is of order $O(N^2)$. Instead, we apply it for each note data $\mathbf{D}_i$, $i = 1, \ldots, 88$ separately which is of order $O(\sum_{i=1}^{88} N_i^2)$.

3 Experiments and Results

We have used the MAPS (MIDI Aligned Piano Sounds) data set [8] in our experiments. The training set is obtained using 440 monophonic piano sounds where we represent the audio by its magnitude spectrogram which is computed via DFT. The spectrogram is formed using overlapping Hanning windows of dimension 2048 with a window shift 512. The final dictionary matrix is 1025×115600 and is around 860 MB. In order to evaluate the performance of the methods we used precision (fraction of retrieved instances that are relevant), recall (fraction of relevant instances that are retrieved) and f-measure $= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$ metrics. The evaluation is performed framewise.

We start with the full solution given in Eq. (2) and obtain an f-measure of 78.07% as in [7]. This performance competes with the leading results obtained on MAPS (81% [8], 77% [1]).

Then, we use exemplar selection methods. We first choose 45 exemplars per note via k-medoids, merge these selected exemplars to form the training data and get an f-measure of 78.69%. Note that whereas selecting columns of $\mathbf{D}$ corresponds to exemplars, selecting certain rows corresponds to frequency selection. Choosing only 400 of the rows via k-medoids in addition to the selected columns and using this skeletonized training data leads to an f-measure of 77.41%. Alternatively, we use affinity propagation. The preference values are set to be a multiple of the median of the similarity values such that nearly 45 exemplars are selected per note. The f-measure of this approach is 78.54%. We also apply affinity propagation on the rows and choose 278 exemplars with an f-measure of 78.39%. That is, we use less than 1% of the training data and still maintain the success rate. In fact, there is a slight increase in the f-measure which supports that removing repetitive and non-relevant parts may lead to better results.

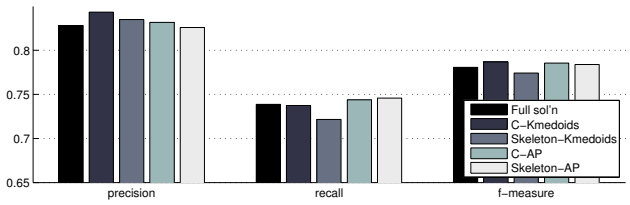

Fig. 1. Results obtained on the test set for all approaches.

The values are obtained by optimizing the threshold on a verification set which excludes the test data. As post-processing, we apply a median filter on each row of $\mathbf{W}$ before thresholding. Filtering leads to an increase around 2-5% in f-measure for all approaches. We have conducted experiments with different polyphony orders and obtained similar results. We observe a minimum of 60% f-measure per polyphony order which shows that the proposed method is robust to polyphony order.

4 Conclusions

We have studied automatic polyphonic music transcription and discussed that the conventional methods are inefficient to handle big data. We show that even a standard matrix factorization model is prohibitive in real applications where a huge amount of training data is used. The update rules are made efficient by the use of exemplar selection techniques. Time and space complexities are improved such that the proposed method can work in real time without compromising the accuracy. A high f-measure value ($\sim$78%) is obtained by using only a few hundred frequency bins and sample columns out of a huge dictionary matrix.

References

1. A. Klapuri and M. Davy, Signal Processing Methods for Music Transcription, Springer, 2006.
2. P. Smaragdis and J. C. Brown, "Non-negative Matrix Factorization for Polyphonic Music Transcription," in *IEEE WASPAA*, pp. 177–180, 2003.
3. P. Smaragdis, "Polyphonic Pitch Tracking by Example," in *IEEE WASPAA*, 2011.
4. H. S. Park and C. H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, Mar. 2009.
5. B. J. Frey and D. Dueck, "Clustering by passing messages between data points.," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
6. D. D. Lee and H. S. Seung, "Learning the Parts of Objects by Non-negative Matrix Factorization," *Nature*, vol. 401, pp. 788–791, 1999.
7. İ. Arı, U. Şimşekli, A. T. Cemgil, and L. Akarun, ""Large Scale Polyphonic Music Transcription Using Randomized Matrix Decompositions," in *EUSIPCO*, 2012.
8. V. Emiya, R. Badeau, and B. 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.