SONG-LEVEL MULTI-PITCH TRACKING BY HEAVILY CONSTRAINED CLUSTERING

Zhiyao Duan, Jinyu Han and Bryan Pardo

Northwestern University
Department of Electrical Engineering and Computer Science
2133 Sheridan Road, Evanston, IL 60208, USA.

ABSTRACT

Given a set of monophonic, harmonic sound sources (e.g. human voices or wind instruments), multi-pitch estimation (MPE) is the task of determining the instantaneous pitches of each source. Multi-pitch tracking (MPT) connects the instantaneous pitch estimates provided by MPE algorithms into pitch trajectories of sources. A trajectory can be short (within a musical note), or long (an entire piece of music). While note-level MPT methods usually utilize local time-frequency proximity of pitches to connect them into a note, song-level MPT is much more difficult and needs more information. This is because pitches evolve discontinuously from note to note, and pitch trajectories can even interweave. In this paper, we cast the song-level MPT problem as a constrained clustering problem. The constraints are time-frequency locality of pitches and the clustering objective is their timbre consistency. Due to this problem's unique properties, existing constrained clustering algorithms cannot be directly applied. We propose a new constrained clustering algorithm. Experiments show that our approach produces good results on real-world music recordings of 4 musical instruments.

Index Terms— Pitch tracking, multi-pitch estimation, fundamental frequency, constrained clustering

1. INTRODUCTION

In an audio mixture of several concurrent harmonic sound sources, estimating and tracking pitches into pitch trajectories for each underlying source is an important problem, called *Multi-pitch Estimation & Tracking*. It has immediate applications to source separation, automatic music transcription, and content-based music search. To date, this problem remains challenging.

Due to its complexity, researchers usually decompose the whole problem into stages: First, they segment an audio example into time frames and estimate pitches in each frame, called *Multi-pitch Estimation (MPE)* [1]. Then, they connect pitch estimates of different frames to form pitch trajectories, called *Multi-pitch Tracking (MPT)*. A pitch trajectory can be short (within a note, *note-level*) or long (goes through the whole song, *song-level*). To address note-level MPT, researchers utilize the local time-frequency proximity of pitches in the same trajectory using different models, e.g. note event models [2, 4] and harmonic temporal structure models [3].

This information, however, is not enough for song-level MPT. This is because pitch trajectories evolve discontinuously from one note to another note and notes are interspersed with silence; pitch trajectories may even interweave. To our knowledge, no existing algorithmic method explicitly addresses the song-level MPT problem.

Song-level MPT is closely related to unsupervised monaural source separation. However, a number of source separation systems [7, 8] are built assuming good multi-pitch trajectories as input. Other are only tested on synthetic data [9], or note-long mixtures [10].

In this paper, we cast the song-level MPT problem as a constrained clustering problem [11, 12, 13], where each pitch trajectory (source) corresponds to a cluster of pitch estimates (*pitches*). Instance-level constraints (must-links and cannot-links) are defined on pairs of pitches, to utilize their local time-frequency locality information. The objective function is defined as the intra-class distance between harmonic structures of pitches, to utilize their timbre consistency. This is reasonable, since humans use timbre consistency as an important cue to help discriminate and track sound sources [5].

According to the definition of our constraints (Section 2.2), our constrained clustering problem has a unique property: almost every pitch estimate is involved in some constraint. This makes existing constrained clustering algorithms inappropriate. In addition, the pitch estimates upon which constraints are applied may not be accurate, making their constraints non-applicable. Therefore, we propose a new constrained clustering algorithm, which minimizes the objective function, while trying to satisfy as many constraints as possible.

The proposed approach is tested on instrumental recordings of ten J. S. Bach four-part chorales. Experimental results are very promising. They also support our claim that both the time-frequency locality (constraints) and the timbre consistency (objective function) is essential to song-level MPT. This paper builds on work in [6] by giving a formal formulation for the song-level MPT problem, introducing a new incremental constrained clustering algorithm for heavily constrained problems.

2. PROBLEM FORMULATION

Given an audio mixture containing K sound sources, and at most K pitch estimates provided by a multi-pitch estimator [14] in each time frame, the song-level MPT problem can be viewed as a pitch clustering problem, where each pitch trajectory (source) is a cluster.

2.1. Timbre consistency

To define the clustering objective function, we note that humans use timbre consistency to discriminate and track sound sources [5]. Thus we try to find a timbre feature for each pitch estimate and make timbre consistency within each pitch trajectory a clustering objective. Since we are dealing with harmonic sources (pitched musical instruments) we represent timbre by *harmonic structure* [9]. This is defined as the vector of relative amplitudes of the harmonics of the pitch, with amplitude measured on a log scale. We expect that different notes produced by the same instrument have harmonic structures

This research was supported in part by NSF grants 0643752 and 0812314.

that are more similar to each other than they are to notes produced by a different instrument [9].

For each pitch estimate in a frame of the audio mixture, we measure the energy in the mixture at first 50 integer multiples of the pitch. This is taken as the harmonic structure of that pitch estimate.

The clustering objective can then be defined as minimizing the intra-class distance of harmonic structures of pitches:

$$f = \sum_{k=1}^{K} \sum_{x_i \in T_k} \|\mathbf{x}_i - \mathbf{c}_k\|^2 \tag{1}$$

where K is the number of pitch trajectories; $\mathbf{x}_i$ is the harmonic structure vector of pitch i; $\mathbf{c}_k$ is the average harmonic structure vector of pitches in trajectory T_k; $\|\cdot\|$ denotes the Euclidean norm.

2.2. Constrained clustering

When minimizing Eq. (1), we do not want to put concurrent pitches into the same cluster, since we assume our sources are monophonic. Also, we prefer to put similar pitches in adjacent frames into the same cluster, since they are likely from the same source. This makes our MPT problem a constrained clustering problem.

Constrained clustering [11, 12, 13] is a class of semi-supervised learning algorithms. Constraints can be imposed on the instance level, where there are two basic forms: must-link and cannot-link. A must-link (cannot-link) specifies that two instances should (should not) be assigned to the same cluster.

In our problem, constraints are imposed on pairs of pitch estimates. A *must-link* is imposed between two pitches in adjacent frames that differ less than 2% in Hz (1/3 of a musical semitone). A *cannot-link* is imposed between two pitches in the same frame. These must-links and cannot-links form the set of all constraints C.

2.3. Properties of the MPT problem formulation

For different audio signals, the thresholds used to define the constraints may vary (e.g. more or less reverberation in a room), but they share two same properties: 1) *Noisy Constraints:* Constraints have errors, since they are defined on pitch estimates which are not always correct. 2) *Heavily Constrained:* Since pitch evolves smoothly over short periods (several frames) and several sources sound simultaneously, nearly every pitch has some must-links and/or cannot-links.

Because of the "Noisy Constraints" property, there may not exist any feasible clustering under all the constraints. This makes existing algorithms [11, 12] inapplicable, since they attempt to to find a clustering minimizing Eq. (1) while satisfying all the constraints.

Instead, we seek an algorithm that minimizes Eq. (1) while satisfying as many constraints as possible. An *Incremental Constrained Clustering* algorithm [13] fits this purpose. It starts from an initial clustering Π_0 that satisfies a subset of all the constraints $C_0 \subset C$ and then incrementally add constraints during following iterations. However, we will show that [13] is inapplicable to our problem in Section 2.4. We will propose a new algorithm in Section 3, which adopts the idea of incremental constrained clustering.

2.4. Forming the initial clustering

For a general incremental constrained clustering problem, the initial clustering Π_0 can be simply set by a random label assignment to all the instances, and the initial constraints C_0 can be set to the empty set $\emptyset$. For our multi-pitch tracking problem, we can give a more informative setting to Π_0 and C_0: We set Π_0 by sorting pitches in each frame from high to low and assigning cluster labels from 1 to K. This is possible because there are at most K pitches in each frame. Also, we define C_0 as the set of all the cannot-links in C, then C_0 will be satisfied by Π_0, because cannot-links are only defined on pitch pairs within a frame and concurrent pitches are in different clusters in Π_0.

Given Π_0 and C_0, we want to minimize Eq. (1) while incrementally adding constraints. Davidson et al. [13] showed that incrementally adding new constraints during clustering is NP-hard in general. But they identified several sufficient conditions under which the clustering could be efficiently updated to satisfy the new and old constraints. The conditions require either (1) at least one instance involved in the new constraint is not currently involved in any old constraint or (2) the new constraint is a cannot-link.

For our MPT problem, however, from the initial constraints C_0, any new constraint from C does not meet any of these sufficient conditions. This is because: 1) due to the "Heavily Constrained" property, almost every pitch estimate has already been constrained by some cannot-links, so Condition 1 is not satisfied. 2) since all the cannot-links are already in C_0, any new constraint will be a must-link, so Condition 2 is not satisfied. Therefore, the algorithm in [13] will do nothing beyond the initial clustering we formed above.

3. NEW CONSTRAINED CLUSTERING ALGORITHM

Here we design a new incremental constrained clustering algorithm, which alternately updates clusterings and constraints from Π_0 and C_0. Suppose we are in the n-th iteration, where the previous clustering is Π_{n-1} and the set of constraints that it satisfies is C_{n-1}, we will not add new constraints according to some "sufficient conditions" as Davidson et al. [13] does. Instead, we first update Π_{n-1} to a new clustering Π_n that also satisfies C_{n-1}, then we expand the set of constraints to C_n, which is the set of all the constraints that can be satisfied by Π_n. So we have $C_{n-1} \subseteq C_n$. Although in some iterations no new constraint may be added, in general the set of satisfied constraints will expand. The algorithm is presented in Algorithm 1. The objective function $f(\Pi)$ decreases when a new clustering is found at Line 2. This part is key to this algorithm and will be explained in Section 3.1. Algorithm 1 terminates when no new clustering can be found at Line 6.

Algorithm 1: IncrementalClustering

```
1  for $n \leftarrow 1$ to $\infty$ do
2      $\Pi_n$ = FindNewClustering($\Pi_{n-1}$,$C_{n-1}$,$f$);
3      if $\Pi_n == \Pi_{n-1}$ then
4          $\Pi' = \Pi_{n-1}$;
5          $C' = C_{n-1}$;
6          return $\Pi'$ and $C'$;
7      else
8          $C_n$ = The set of all constraints satisfied by $\Pi_n$;
9      end
10 end
```

3.1. Find a new clustering by swapping labels

In Line 2 of Algorithm 1, we want to update Π_{n-1} to a new clustering Π_n that also satisfies C_{n-1}. We do this by moving at least one point between clusters in Π_{n-1}. However, if we move some point p from cluster T_k to cluster T_l, all the points that have a must-link to p according to C_{n-1} should be moved from T_k to T_l. Then all the points in cluster T_l that have cannot-links to either above-mentioned point need also be moved out of T_l. This may cause a chain reaction.

Here we define an operation *swap* to change the label of p. This operation swaps cluster labels of all the points in the *swap set* of node p between clusters T_k and T_l, which is defined as the set of points in these clusters that form a connected graph containing p, where each node is a point and each edge is a must-link or cannot-link. Figure 1 illustrates a swap set. It is easy to see that after a swap operation, the newly found clustering satisfies all the previous constraints. This is because the swap set is isolated from the other part of the graph, and all constraints inside are maintained.

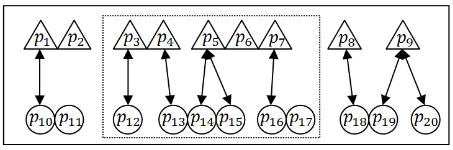

Fig. 1. An illustration of the swap set (dashed box) of node p_{15} between the triangle cluster and the circle cluster. Must-links are represented by two nodes touching (e.g. between p_1 and p_2); cannot-links are represented as arrows.

In the "FindNewClustering" function of Algorithm 1, we randomly traverse all the points and try the swap operation until we find a new clustering Π_n that decreases the objective function. This subroutine is described in Algorithm 2.

Algorithm 2: FindNewClustering

```
1  for Randomly traverse all the points $p_1, \cdots, p_M$ do
2      Suppose $p_m$ is in cluster $T_k$;
3      $J_{best} = f(\Pi_{n-1})$;
4      for $l \leftarrow 1, \cdots, K; l \neq k$ do
5          Find the swap set of point $p_n$ between $T_k$ and $T_l$;
6          From $\Pi_{n-1}$, do swap to get a new clustering $\Pi_s$;
7          if $f(\Pi_s) < J_{best}$ then
8              $J_{best} = f(\Pi_s)$;
9              $\Pi_n = \Pi_s$;
10         end
11     end
12     if $J_{best} < f(\Pi_{old})$ then
13         return $\Pi_n$;
14     end
15 end
16 return $\Pi_{n-1}$;
```

3.2. Discussion

Given the number of all points M and the number of clusters K, the running time of each iteration of Algorithm 1 is $O(KM^2)$. This is because in Algorithm 2, there are MK nested loops from Line 5 to Line 10. Line 5, 6 and 8 all cost $O(M)$ operations. In addition, Algorithm 1 always terminates, because the space of feasible solutions is finite and in every iteration the new clustering found by "FindNewClustering" monotonically decrease the objective function.

However, we do not know how many iterations Algorithm 1 may take analytically. In the experiments of our multi-pitch tracking problem, where there are about 12,000 points and 20,000 pairwise constraints, the algorithm terminates in hundreds of iterations.

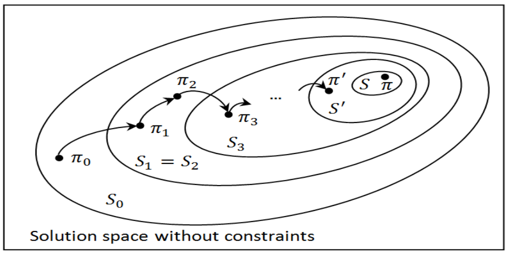

Fig. 2. An illustration of Algorithm 1. Ellipses represent solution spaces under constraints in different iterations. Points represent clusterings. Arrows show how clusterings are updated to decrease the objective function.

Figure 2 illustrates the process of Algorithm 1 from the perspective of solution spaces. The algorithm starts with the initial constraints C_0 and clustering Π_0, where the solution space under C_0 is S_0. Then it updates to a new clustering Π_1 in S_0 which decreases the objective function f. After adding all the new constraints that Π_1 satisfies, the set of satisfied constraints is expanded C_1, and the solution space is shrunk to S_1. Then, a new clustering Π_2 is updated in S_1, but this time there is no new constraint satisfied. Therefore, $C_2 = C_1$ and $S_2 = S_1$. This iteration terminates in Π' and C', where Π' is a local minimum of f in the solution space S'. S is the solution space under all the constraints C, and Π is its optimal solution. It is noted that if the constraints are noisy, S might be empty.

4. EXPERIMENT

4.1. Data set and error measure

The data set we used was ten pieces of J.S. Bach four-part chorales, totalling 330 seconds of audio. Each piece was performed by a quartet of instruments: violin (Track 1), clarinet (Track 2), tenor saxophone (Track 3) and bassoon (Track 4). Each musician's part was recorded in isolation as 44.1 kHz, 16 bit PCM audio. These individual recordings were then mixed together into single-channel recordings containing all four parts.

Each recording was broken into 46 ms frames with 10 ms between frame centers. Ground-truth pitch trajectories for each piece were created using a robust single pitch detection algorithm [15] on the isolated instrument recordings prior to mixing the recordings together. Ground-truth pitch tracks were then manually corrected.

Pitch estimates for each frame were obtained with our previously published multi-pitch estimator [14]. Pitch tracks were derived from pitch estimates using the approach described in this paper.

We evaluate the proposed approach at the frame-level. For each estimated pitch trajectory, a pitch estimate is called correct if it deviates less than 3% in Hz (a quarter-tone) from the pitch in the same frame in the ground-truth pitch trajectory. This threshold is in accordance with the standard tolerance used in measuring correctness of pitch estimation for music [1]. Then accuracy is calculated for each pitch trajectory of each piece of music as Acc $= \frac{\text{TP}}{\text{TP+FP+FN}}$, where TP (true positives) is the number of correctly clustered pitches, FP (false positives) is the number of pitches that do not belong to but are clustered to the trajectory, and FN (false negatives) is the number of pitches that belong to but are not clustered to the trajectory.

4.2. Results

As there is no existing method addressing the song-level MPT problem, we investigate the effectiveness of different techniques that are used in our method. The proposed algorithm utilizes both the time-frequency locality information (represented by constraints in Section 2.2) and the timbre consistency information (represented by the objective function in Section 2.1). As claimed in Section 1, both are necessary for song-level MPT. In order to show this, we run two baseline iterative algorithms: one called “Constraints only” tries to satisfy as many constraints as possible while ignoring the objective function; the other called “Objective only” tries to minimize the objective function while ignoring the constraints. Both algorithms start from the same initial clustering (denoted as “Initial”) as the proposed algorithm, which is obtained simply according to pitch heights in each frame, as described in Section 2.4.

Figure 3 shows box plots of MPT accuracy comparisons. Each box consists of 40 points, corresponding to 4 tracks of 10 music pieces. The lower and upper lines of each box show 25th and 75th percentiles of the sample. The line in the middle is the sample median, which is also presented as the number in each box. The lines extending above and below each box show the extent of samples, excluding outliers. Outliers are defined as points over 1.5 times the interquartile range from the sample median and are shown as crosses.

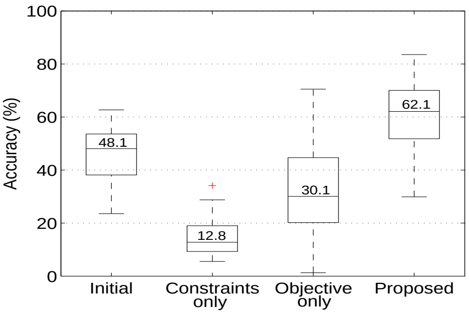

Fig. 3. Song-level MPT accuracy comparisons.

Compared to the initial clustering, both “Constraints only” and “Objective only” significantly reduce the MPT accuracy. However, utilizing both together, the proposed method significantly improves the accuracy. This supports our claim that song-level MPT requires both the time-frequency locality information and the timbre consistency information of pitches.

Although the median accuracy of the proposed algorithm is only 62.1%, we note that the input to our algorithm is the multi-pitch estimation (MPE) results provided by [14], which are not error-free. A wrong pitch estimate cannot be correctly put into any pitch trajectory no matter what algorithm is used. In [14], the average MPE accuracy among all 10 pieces is 70.0±3.1%. If we assume these MPE errors are evenly distributed into all the tracks, then it is the upper bound accuracy for any song-level MPT algorithm that works on these pitch estimates. In fact, among all the correct pitch estimates provided by [9], on average 89.1% of them in each track are put into the correct pitch trajectory by the proposed algorithm. For a random guess, however, only 25% of correct input pitch estimates can be put into the correct pitch trajectory. In this sense, the proposed algorithm obtains very good results.

5. CONCLUSION

In this paper, we propose an approach to address the song-level multi-pitch tracking (MPT) problem, which no existing method explicitly addresses. We claim that both the time-frequency locality information and the timbre consistency information of input pitch estimates should be utilized. We cast the problem as a constrained clustering problem, where constraints represent the former information and the objective function represents the latter. The unique characteristics of our problem makes previous constrained clustering algorithms inapplicable. Therefore, we design a new algorithm that minimizes an objective function while satisfying as many constraints as possible. Experiments on 10 pieces of recorded music demonstrate good performance.

6. REFERENCES

[1] A. Klapuri, “Multiple fundamental frequency estimation by summing harmonic amplitudes,” *Proc. ISMIR*, pp. 216-221, 2006.

[2] M. Ryynänen and A. Klapuri, “Polyphonic music transcription using note event modeling,” *Proc. WASPAA*, pp. 319-322, 2005.

[3] H. Kameoka, T. Nishimoto, and S. Sagayama, “A multipitch analyzer based on harmonic temporal structured clustering,” *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 15, No. 3, pp. 982-994, 2007.

[4] W.-C. Chang, A. W.Y. Su, C. Yeh, A. Roebel and X. Rodet, “Multiple-F0 tracking based on a high-order HMM model,” *Proc. DAFx*, 2008.

[5] A. S. Bregman, *Auditory Scene Analysis*, Cambridge, MA, MIT Press, 1990.

[6] Z. Duan, J. Han and B. Pardo, “Harmonically informed multi-pitch tracking,” *Proc. ISMIR*, 2009.

[7] T. Virtanen and A. Klapuri, “Separation of harmonic sounds using multipitch analysis and iterative parameter estimation,” *Proc. WASPAA*, pp. 83-86, 2001.

[8] Y. Li, J. Woodruff and D.-L. Wang, “Monaural musical sound separation based on pitch and common amplitude modulation,” *IEEE Trans. Audio Speech Language Process.*, Vol. 17, No. 7, pp. 1361-1371, 2009.

[9] Z. Duan, Y. Zhang, C. Zhang and Z. Shi, “Unsupervised single-channel music source separation by average harmonic structure modeling,” *IEEE Trans. Audio Speech Language Process.*, Vol. 16, No. 4, pp. 766-778, 2008.

[10] M. Lagrange and G. Tzanetakis, “Sound source tracking and formation using normalized cuts,” *Proc. ICASSP*, pp. 61-64, 2007.

[11] K. Wagstaff and C. Cardie, “Clustering with instance-level constraints,” *Proc. ICML*, pp. 1103-1110, 2000.

[12] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, “Constrained K-means clustering with background knowledge,” *Proc. ICML*, pp. 577-584, 2001.

[13] I. Davidson, S. S. Ravi and M. Ester, “Efficient incremental constrained clustering” *Proc. ACM SIGKDD*, pp. 240-249, 2007.

[14] Z. Duan, B. Pardo and C. Zhang, “Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions,” under view.

[15] A. de Cheveigné and H. Kawahara, “YIN, a fundamental frequency estimator for speech and music,” *J. Acoust. Soc. Am.*, Vol. 111, pp. 1917-1930, 2002.