Citation for published version


DOI

Link to record in KAR

http://kar.kent.ac.uk/42367/

Document Version

UNSPECIFIED

Copyright & reuse
Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (e.g., Creative Commons), permissions for further reuse of content should be sought from the publisher, author, or other copyright holder.

Versions of research
The version in the Kent Academic Repository may differ from the final published version. Users are advised to check http://kar.kent.ac.uk for the status of the paper. Users should always cite the published version of record.

Enquiries
For any further enquiries regarding the licence status of this document, please contact: researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at http://kar.kent.ac.uk/contact.html
ABSTRACT

Much polyphonic music is constructed from several melodic lines - known as voices - woven together. Identifying these constituent voices is useful for musicological analysis and music information retrieval; however, this voice-identification process is time-consuming for humans to carry out. Computational solutions have been proposed which automate voice segregation, but these rely heavily on human musical knowledge being encoded into the system. In this paper, a system is presented which is able to learn how to separate such polyphonic music into its individual parts. This system uses a training corpus of several similar pieces of music, in symbolic format (MIDI). It examines the note pitches in the training examples to make observations about the voice structures. Quantitative evaluation was carried out using 3-fold validation, a standard data mining evaluation method. This system offers a solution to this complex problem, with a 12% improvement in performance compared to a baseline algorithm. It achieves an equal standard of performance to heuristic-based systems using simple statistical observations: demonstrating the power of applying data-driven techniques to the voice separation problem.

1. INTRODUCTION

A common compositional device in music is to construct a piece by interweaving several melodic lines. In musicology, these melodic lines are often referred to as voices [3]. Each voice can be considered independently as a melodic pattern which is complete and interesting in its own right. Several related voices, combined together to form one piece of polyphonic music, can generate additional harmonic qualities to enhance the voices.

Fugues provide a perfect example of this compositional technique in action, being constructed solely of a number of different melodic voices. J. S. Bach was a fundamentally important composer in the history of fugue composition; in particular his highly influential work The Well-Tempered Clavier comprises 48 fugues.

In analysis of music such as Bach fugues, the musicologist identifies individual voices to facilitate more advanced analysis of the melodic content such as finding thematic patterns common to different voices. The musical score usually gives the musicologist much help in identifying each voice, as each voice is notated slightly differently (the direction of the note stems often indicates which voice each note belongs to).

Identifying each voice is a considerably harder task if these notational clues are not present. In such cases the musicologist needs to examine musical detail within the piece, such as the note pitches and rhythms [7, 12], using this in conjunction with their knowledge of voice structure in that compositional style. Voice identification can often be a painstaking and time-consuming process.

Can a computer learn how to extract the constituent voices from a piece of music by similar examination of musical detail? Given minimal human assistance and a training set of similar music with the voices already identified, I propose that patterns of voice movement can be identified and learnt. Such patterns represent learned knowledge of compositional style that can assist the computer in identifying individual voices in other music of a similar style. Computers should complete this task markedly faster than a human, due to processing speeds available.

2. RELATED WORK

In recent years a number of different solutions have been proposed for the task of voice segregation [2, 4, 8, 9, 10, 11]. Prior to 2004, voice identification was considered supplementary to the primary task of transcription to notation from musical input [2, 9] but has recently become a problem of interest in its own right [3, 4, 8, 10, 11].

Previous work has imposed human musical knowledge on the system in the form of rules and heuristics [2, 4, 8, 9, 11], rather than enabling the system to learn how voices are structured. In other words, the program is told exactly how to solve the voice segregation problem rather than allowing it to learn how to piece together the voices. These pieces of work use perceptually-motivated rules [7, 12] and have been successful to a certain degree; however, the computer is not learning these rules and developing its own knowledge, but merely utilising the knowledge provided by human investigation. Higher-level voice-leading principles provide a heuristic guide to typical routes that voices take throughout the course of the piece. For extreme styles of music, these principles may fail, particularly in more contemporary music that challenges the rules.
The system was implemented in Matlab, making use of the MIDI toolbox for Matlab [6] to process MIDI files containing the training and test corpora. A MIDI file is returned by the system such that the MIDI channel marks the voice that each note belongs to. The lowest voice is in channel 1, the second lowest in channel 2 and so on. The system learns from a training corpora of music files, examining how likely each possible MIDI pitch is to occur for each voice, and how likely each transition between pitches is to occur.

The voice identification algorithm for a given piece is:

PRE-PROCESSING: Transpose the piece into the normalised key of C so that the training data is not skewed harmonically

STEP 1: Find marker points: points in time where each voice is present and the pitches are distributed far apart

STEP 2: For each marker point: define windows centred around each marker point, which extend out to meet halfway between each marker point (see Figure 2)

STEP 3: For each window: work outwards from the marker to the window edges. Allocate each note to a voice using the probabilities learnt in training to maximise the cost function:

$$\max_v [P(V(x) = v | V(n) = v) + (0.5 \times P(V(x) = v))]$$ (1)

where $V(x)$ is the voice allocated to note $x$ and note $n$ is the previous note in voice $v$.

If at any point, more than one synchronous note is allocated to one voice, give priority to the highest scoring note

STEP 4: Similarly allocate voices to the notes from the first marker, backwards, to the beginning of the piece

STEP 5: Similarly allocate voices to the notes from the last marker, forwards, to the end of the piece

POST-PROCESSING: Transpose the fugue back to its original key (i.e. reverse the pre-processing step)

3.2. Implementation Details

The voice identification algorithm for a given piece is:

PRE-PROCESSING: Transpose the piece into the normalised key of C so that the training data is not skewed harmonically

STEP 1: Find marker points: points in time where each voice is present and the pitches are distributed far apart

STEP 2: For each marker point: define windows centred around each marker point, which extend out to meet halfway between each marker point (see Figure 2)

STEP 3: For each window: work outwards from the marker to the window edges. Allocate each note to a voice using the probabilities learnt in training to maximise the cost function:

$$\max_v [P(V(x) = v | V(n) = v) + (0.5 \times P(V(x) = v))]$$ (1)

where $V(x)$ is the voice allocated to note $x$ and note $n$ is the previous note in voice $v$.

If at any point, more than one synchronous note is allocated to one voice, give priority to the highest scoring note

STEP 4: Similarly allocate voices to the notes from the first marker, backwards, to the beginning of the piece

STEP 5: Similarly allocate voices to the notes from the last marker, forwards, to the end of the piece

POST-PROCESSING: Transpose the fugue back to its original key (i.e. reverse the pre-processing step)

All MIDI files were sourced from the Humdrum database, available at http://kern.humdrum.net (accessed January 2008)
4. EVALUATION

Quantitative evaluation was carried out using 3-fold validation [13]. Performance was measured using standard information retrieval statistics: precision (the percentage of notes allocated to a voice that correctly belong to that voice), recall (the percentage of notes in the voice that are successfully allocated to that voice) and F-measure (which reflects a balance of precision and recall). The system was compared to a baseline algorithm that used pitch ordering for allocation: at each timepoint, it allocated the lowest note to the lowest voice, the next lowest note to the next lowest voice and so on. If less notes are sounding than voices (i.e. one of the voices is silent), then it allocated voices from the bottom voice up, with upper voices unallocated (silent). Figure 3 shows this.

J. S. Bach’s famous collection of fugues, The Well Tempered Clavier, supplied the corpus for the first experimentation. All 26 three-voice fugues were tested (see Table 1), as were all 19 four-voice fugues (Table 2) from this set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Voice</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>My system</td>
<td>v3</td>
<td>90.53%</td>
<td>88.84%</td>
<td>89.48%</td>
</tr>
<tr>
<td>My system</td>
<td>v2</td>
<td>82.92%</td>
<td>83.40%</td>
<td>83.01%</td>
</tr>
<tr>
<td>My system</td>
<td>v1</td>
<td>92.44%</td>
<td>92.80%</td>
<td>92.44%</td>
</tr>
<tr>
<td>Baseline</td>
<td>v3</td>
<td>97.27%</td>
<td>63.28%</td>
<td>76.01%</td>
</tr>
<tr>
<td>Baseline</td>
<td>v2</td>
<td>67.62%</td>
<td>74.89%</td>
<td>70.80%</td>
</tr>
<tr>
<td>Baseline</td>
<td>v1</td>
<td>80.09%</td>
<td>99.15%</td>
<td>88.37%</td>
</tr>
</tbody>
</table>

Table 1. Voice identification: Bach three-voice fugues. Voice 3 is the highest voice and voice 1 is the lowest voice.

An interesting but under-explored avenue in previous work is how systems perform on other styles of music. So the second set of training data consisted of all 17 string quartets composed by Beethoven (see Table 3).

<table>
<thead>
<tr>
<th>Method</th>
<th>Voice</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>My system</td>
<td>violin1</td>
<td>79.84%</td>
<td>69.47%</td>
<td>71.86%</td>
</tr>
<tr>
<td>My system</td>
<td>violin2</td>
<td>59.79%</td>
<td>57.91%</td>
<td>58.68%</td>
</tr>
<tr>
<td>My system</td>
<td>viola</td>
<td>60.86%</td>
<td>59.38%</td>
<td>60.00%</td>
</tr>
<tr>
<td>My system</td>
<td>cello</td>
<td>71.55%</td>
<td>72.02%</td>
<td>71.70%</td>
</tr>
<tr>
<td>Baseline</td>
<td>violin1</td>
<td>80.08%</td>
<td>51.41%</td>
<td>62.07%</td>
</tr>
<tr>
<td>Baseline</td>
<td>violin2</td>
<td>64.39%</td>
<td>57.03%</td>
<td>60.29%</td>
</tr>
<tr>
<td>Baseline</td>
<td>viola</td>
<td>63.52%</td>
<td>67.07%</td>
<td>65.06%</td>
</tr>
<tr>
<td>Baseline</td>
<td>cello</td>
<td>66.91%</td>
<td>90.68%</td>
<td>76.54%</td>
</tr>
</tbody>
</table>

Table 2. Voice identification in Bach four-voice fugues. Again voice 1 is the lowest voice, voice 4 the highest.

<table>
<thead>
<tr>
<th>Method</th>
<th>Voice</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>My system</td>
<td>violin1</td>
<td>80.88%</td>
<td>80.85%</td>
<td>80.86%</td>
</tr>
<tr>
<td>My system</td>
<td>violin2</td>
<td>88.65%</td>
<td>65.57%</td>
<td>75.38%</td>
</tr>
<tr>
<td>My system</td>
<td>viola</td>
<td>95.94%</td>
<td>70.11%</td>
<td>81.02%</td>
</tr>
<tr>
<td>My system</td>
<td>cello</td>
<td>93.19%</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Table 3. Voice identification: Beethoven String Quartets.

5. DISCUSSION OF RESULTS

My system performed noticeably better than the baseline algorithm at identifying each voice in Bach’s fugues, averaging 80.5% F-measure compared to the baseline’s 68.7%. Reasonable F-measures (average 64.4%) were also recorded when identifying each instrumental part in the Beethoven string quartets.

The system performed more strongly on the Bach fugues than on the Beethoven music. This is likely to be due to the greater variety in compositional style between differ-

---

3 In 3-fold validation, the training corpus is divided into three sets. Over three training runs, two sets are used for training and one for testing, till all three sets have been used as test data.

4 Evaluation of voice identification systems is non-trivial and worthy of discussion in its own right at greater length; however for ease of comparison between different systems I adopt this strategy for now.

5 String quartet music occasionally requires instruments to play more than one note simultaneously; however in general each piece in the corpus was separable into almost entirely monophonic vocings.

6 The design of the baseline algorithm clearly favoured recall in lower voices and precision in the higher voices. However the average F-measure score reflects a balance of overall precision and recall.
ent Beethoven string quartet compositions, compared to Bach fugues which have a more uniform style. Voice separation in Beethoven string quartets was expected to be a harder task than for Bach fugues. Evaluation showed this to be the case, with lower F-measure, precision and recall scores. With average F-measure scores of over 60% for voice segregation in the Beethoven corpus, though, it is pleasing to see some potential in how the system generalises to work on music other than Bach fugues; an under-explored aspect in previous work.

The functionality of this system requires the existence of a set of relevant training examples. This is synonymous with a human needing experience of similar music before attempting to identify the voices in a new piece of music. If pieces by that composer, of a similar genre, do not exist for training, then more general training examples can be supplied to the system and the system will still be able to make a reasonable attempt at voice segregation. To illustrate this, voice separation carried out on Mozart’s Fugue in C minor, (K. 546: mvt. 2), using Bach’s fugues to train on, scored a mean F-measure of 75%.

This voice segregation system matches the standards of previous systems, despite having less human knowledge encoded in its operation. Its learning approach produced similar results to the systems driven by human-devised heuristics [4, 8, 11] (as demonstrated by the F-measure scores in Table 4). Better results were achieved than for VoiSe [10], the only other system incorporating learning; though a fair comparison cannot be made until VoiSe is tested on a comparable repertoire to the other systems.

6. FUTURE WORK

Reliance on human knowledge, although minimised compared to other systems, is still present in the system presented here. For this system to demonstrate artificial intelligence further, the data mining approach could be increased substantially. More observations from the training data could be incorporated, such as timing information, or observations about the nature of marker points. Ideally the system would identify for itself what is musically important for tracing the route of the voices through the piece.

More complex statistical tools could also be utilised. Currently the system only considers a history of one note previous to the current note, when allocating the current note to a voice. It would be interesting to apply Hidden Markov Models here, so that more of the previously allocated notes can be used to assist in voice allocation.

This paper has focussed exclusively on finding voices exactly as the composer has written them. However the written voiceings do not always correspond to the melodic lines that we perceive when listening to music [1, 3, 5]. This system could be used with good effect to detect such higher-level voices, given appropriate training examples.

7. CONCLUSIONS

Machine learning has provided a solution to the problem of dividing polyphonic music into its individual voices. From a small set of observations from training data, the system presented in this paper can identify the route that a voice takes through a piece. It is able to identify constituent voices of a polyphonic piece of music with good precision and recall; an average F-measure of 80% was recorded for Bach fugues and 64% for Beethoven string quartets (which vary more in style than Bach fugues).

Performance in the fugal voice-separation task was equal to that of more heuristically guided systems [8, 11] and surpassed that of a baseline algorithm which allocated voices purely on relative pitch positioning. While improvements could be made to this system to enhance the knowledge it gains from data mining and further minimise the human knowledge it uses, this work represents an advance in the application of computational methods to the voice separation problem. It offers an alternative approach to that of encoding human-imposed heuristics and rules. There is much potential for further exploration of artificial intelligence techniques to the voice separation problem and to music analysis in general.

8. ACKNOWLEDGEMENTS

This work has benefitted from comments from Nick Collins, Chris Thornton, Chris Darwin and three reviewers.

9. REFERENCES