A SIMILARITY ALGORITHM FOR INTERACTIVE STYLE IMITATION

Aengus Martin, Craig T. Jin, Alistair McEwan
School of Electrical and Information Engineering
Sydney University, NSW 2006, Australia
aengus.martin@sydney.edu.au

William L. Martens
Faculty of Architecture, Design and Planning
Sydney University, NSW 2006, Australia

ABSTRACT

We consider an interactive musical agent (IMA) which shares control of an ensemble of virtual instruments with a human performer, and which is required to interact in a particular musical style without a priori knowledge of the ensemble itself. We present an algorithm which allows such an agent to find the segment in a corpus of demonstration performances, which best matches a given musical situation. This segment can then be used to choose values for the parameters under the agent's control. We show the algorithm's efficacy by demonstrating an IMA which can (i) perform alongside a human musician and (ii) imitate the behaviour of another artificial musical agent which performs according to a set of rules.

1. INTRODUCTION

An interactive musical agent (IMA) is a software entity designed to play music alongside a human performer. In a *shared control* context [2], the IMA and the human performer share control of an ensemble of virtual instruments. The design of IMAs is an ongoing research area and a wide range of approaches have been investigated, including complex combinations of heuristic and stochastic processes [6], artificial neural-network based learning systems [3], partially observable Markov decision processes [7], and many others. In some cases, IMA development has been combined with research into musical style imitation, notably in the *Omax* system of Assayag and Dubnov [1] and Pachet's *Continuator* [8].

The Continuator stores a *demonstration data set* comprising symbolic representations of music performed in a particular style. During an interactive performance, it finds the segment in the demonstration data set which best matches the current musical note and a variable-length history of previous notes, in order to choose the next note. When it cannot find a match for the current musical situation in the demonstration data, it uses musically-informed approximate representations of the sequence of notes. For example, if an initial search finds no match for a sequence of notes described by their pitch, duration and velocity attributes, a second search can be conducted using only approximate pitch and velocity attributes, since these provide the most musically salient approximation of the note sequence [8]. This is a key feature of the Continuator and it allows the system's performance to "gracefully degrade" as the musical situation deviates from those found in its training data. Similar ideas are used in Omax.

The aim of our research is to adapt the interactive style imitation approach used in these systems, to the design of an IMA which shares control of an ensemble of virtual instruments with a human performer. Specifically, we consider an ensemble which is controlled at a relatively high level, using switches which turn voices and algorithmic processes on and off, rather than at the lower level of notes or gestures. In addition, we address the scenario in which the IMA has no a priori knowledge of the ensemble; it does not know what the switches control. This scenario contrasts with those for which Omax and the Continuator were designed, because the IMA cannot make musically-informed approximations of the parameter control data–it has no information on which to base such approximations. A new method is required to find matches in the demonstration data set. To this end, we identified an algorithm from the field of data mining called the *similarity measure for sequential patterns* (S^2MP) [9]. It is used to quantitatively describe the similarity between two sequences of sets of integers (i.e. sequences in which each element is a set of integers; *set sequences*, for convenience).

In this paper, we present an IMA for use in a shared control context. At regular instants during a performance, it uses the S^2MP algorithm to find the best match in a demonstration data set in order to choose values for the parameters under its control. We show its effectiveness in two ways. First, we demonstrate its real-time use, sharing control of an ensemble of virtual instruments with a human musician. Second, we show that it is able to imitate the behaviour of another artificial musical agent which performs according to a known set of rules.

2. AN IMA BASED ON SIMILARITY

Our IMA was designed for an interactive music scenario in which it shares control of an ensemble of virtual instruments with a human performer. The ensemble is controlled by a set of switches which are used to turn voices and algorithmic processes on and off. Each switch corresponds to a parameter which can take the value 1 (on) or zero (off). A set, M, of these parameters is controlled by the musician and the remaining ones, C, are controlled by the IMA. At regular *decision times* during a performance, t_i ($i = 1, 2, 3, \ldots$), the IMA updates the values of the parameters in C. At time t_i, its choice of parameter values are based on (i) a stored demonstration data set, $\mathfrak{D}$, comprising N-example performances, $\mathfrak{D}^1, \ldots, \mathfrak{D}^N$, and (ii) the parameter values used in the performance so far. This includes both musician- and IMA-controlled parameters up to time t_{i-1} and the values of the musician-controlled parameters at time t_i just before the IMA chooses new values for its parameters. Each example in the demonstration data set is encoded as a set sequence. This is done by encoding the parameter values at each decision time as a set of integers indicating which switches were turned on at that time.

The new parameter settings are chosen as follows. At time t_i, the IMA forms a *performance set sequence*, $\mathscr{P}$, from the parameter values at the last K-decision points (i.e. t_{i-K+1}, ..., t_i; K is a positive integer) in the performance data. It does this by forming a set sequence in which the jth set ($j = 1, \ldots, K$) encodes the set of switches that were turned on at t_{i-K+j}. The IMA then iterates over all length-K set sequences in the demonstration data and for each one, it calculates a similarity score using the S^2MP algorithm (see below). Since values for the IMA-controlled parameters have not yet been chosen at t_i, the final set in $\mathscr{P}$ is incomplete. Therefore, each time the similarity is calculated between a set sequence, $\mathscr{Q}$, from $\mathfrak{D}$, and the performance set sequence, $\mathscr{P}$, the last set of $\mathscr{P}$ is modified to maximise the similarity with $\mathscr{Q}$, i.e. it is modified so that the IMA-controlled parameter values encoded by its final set (that corresponding to t_i) are the same as those encoded by the last set of $\mathscr{Q}$. The best-matching set sequence, $\mathscr{Q}^*$, is that corresponding to the highest similarity score. Once $\mathscr{Q}^*$ has been found, the IMA extracts the values of the IMA-controlled parameters from its final set and uses these values at t_i. A mathematical description of this procedure is given in the Appendix.

The S^2MP quantitatively describes the similarity between two set sequences. We give a brief outline here (see [9] for details). Given two set sequences a *similarity score* is calculated. This is a value between 0 and 1, where 1 indicates that the set sequences are identical, and 0 indicates that no set from the first set sequence has any elements in common with any set from the second set sequence. It is a weighted average of two measures of similarity: a *mapping score* and an *order score*. The mapping score describes the average similarity between individual sets in the two set sequences. The order score describes the extent to which the set sequences are ordered in the same way.

The IMA was implemented in software as a new object for Max[1] and will be made available on request from the authors. In the following, we describe its evaluation in real-time interaction with a human musician, and in a simulated interaction with an artificial, rule-based musician.

3. EVALUATION

3.1. Real-time interaction with a musician

In the first evaluation, the IMA interacted with a human musician in a shared control scenario. An ensemble of virtual instruments was created in Max. It comprised ten instruments which were synchronised in tempo and harmony by a central system, and controlled at a note-level by various algorithmic processes. The user-interface comprised ten toggle switches (see Figure 1) which could be used to switch on and off the instruments, so the musician's creative role is like that of a musical arranger, deciding which instruments should play and when. Our aim was to investigate if the IMA would act (i) to use combinations of instruments consistent with the demonstration data and (ii) to control the instruments consistent with long term structure in the performance- e.g. would it successfully differentiate between a build-up and a reduction of intensity. To begin, we created a demonstration data set comprising recordings of the control data from six performances (see Figure 2a). Then a performance was conducted with the IMA controlling five switches and a musician controlling the other five. Observations made during this part of the evaluation are given in the next section.

Figure 1. Screenshot of interface. The superimposed white circles indicate the toggle switches which can be controlled by a musician or the IMA.

3.2. Simulated interaction

The second evaluation was done by creating "artificial musicians" with which the IMA could interact. Two artificial musicians were created (H1 and H2), each of which operated according to a simple set of rules. Our aim was to investigate if the IMA could use a demonstration data set created by an artificial musician to act consistently with the underlying rules. The behavioural rules were in the form of dependencies between parameters. The rules associated with H1 are indicated in Figure 2b (the set of rules for H2 are not shown but had a similar level of complexity). In this figure, the nodes of the tree represent parameters and arrows indicate dependencies. Arrows with black circles indicate inverse dependencies (e.g. ② $\rightarrow$ 1 causes ④ $\rightarrow$ 0) and the numbers indicate probabilistic dependencies (i.e.

[1] www.cycling74.com

the probability that a change will be induced is less than unity). Finally changes induced may occur after a delay of 0 or 1 or 2 time steps. The number of time steps in each case is chosen randomly.

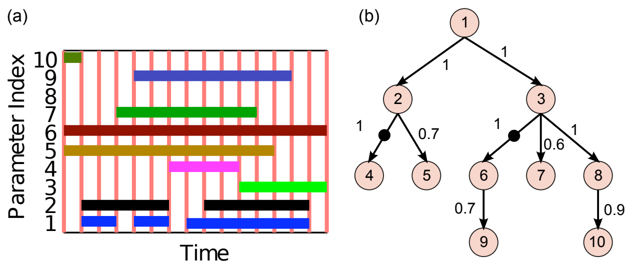

Figure 2. (a) An example from the demonstration data set used in the first evaluation. Horizontal lines indicate when a parameter was set to 1. (b) A tree indicated the rules underlying one of the synthetic musicians used in the second evaluation.

For each trial in this quantitative evaluation, an artificial musician was used to generate a set of demonstration examples. Then an interaction was simulated between our IMA and the artificial musician. Performance was measured by calculating the percentage of times that a required parameter change was not present (as either a cause or effect), as a fraction of the total number of changes in the interaction. This percentage is denoted p_b.

The experimental conditions were chosen to investigate the effect on style emulation accuracy of (i) the number of training examples, N, (ii) K, the history-length to use when searching the demonstration data, and (iii) the choice of artificial musician, H. As a post-hoc analysis, we studied the effect of C, the choice of parameters under IMA control. Specifically, we used the following four conditions: $\{K = 4, C = \{6,7,8,9,10\}, H = H1\}$, $\{K = 4, C = \{2,3,4,5,6\}, H = H1\}$, $\{K = 7, C = \{6,7,8,9,10\}, H = H1\}$, $\{K = 4, C = \{2,4,6,7,9\}, H = H2\}$. For each condition we ran the experiment four times with different numbers of training examples, $N = \{2,6,10,20\}$.

4. RESULTS

The observations made by the musician after real-time interaction with the IMA were as follows. First, the IMA generally avoided combinations of instruments which were not found in the dataset. However, occasionally it changed the value of a parameter two or more times in subsequent time steps when no such rapid changes were found in the demonstration data. We found that this occurred when the matching algorithm switched between two different areas of the demonstration data from one time step to the next. Second, the IMA usually differentiated successfully between long term structures such as gradual, build-ups and break-downs (there were examples of both in the demonstration data set).

The results of the interaction simulation experiments are summarised in Figure 3. The experimental conditions are shown in the legend. In general, accuracy increases (p_b decreases) as the number of demonstration examples is raised. Two exceptions to this are conditions 2 and 3 where the mean p_b value increases from $N = 10$ to $N = 20$. However the error bars are quite large for these conditions.

Conditions 1 and 2 differ only in the selection of parameters that the IMA is controlling. We note that for every link to and from a parameter node in a heuristic tree, a style-rule must be satisfied. The total number of links to and from the nodes in Figure 2b associated with the parameters in condition 1, is 5 where as that for condition 2 is 8. Therefore, condition 2 presents a more complex style-emulation task and this is reflected in higher p_b-values.

The large error bars indicate that performance varied considerably across trials. This is partly due to the fact that the demonstration data was randomly generated for each trial. In some cases the demonstration data was not a good representation of the variety of output that a given heuristic scheme could produce. This happened more frequently with lower values of N, and this explains why the error bars are larger for the $N = 2$ conditions.

In condition 1, $K = 4$, whereas in condition 2, $K = 7$. Performance is better in condition 1. We suspect that this is because $K = 4$ is close to optimum for the heuristic schemes used, since parameter changes induce effects which always occur with in two time steps.

For all of the experiments we qualitatively analysed the errors made by the IMA. We observed that when a rule was broken, it was usually because parameters were changed one or two time steps too early or too late, rather than because the wrong parameter was changed or no parameter was changed at all.

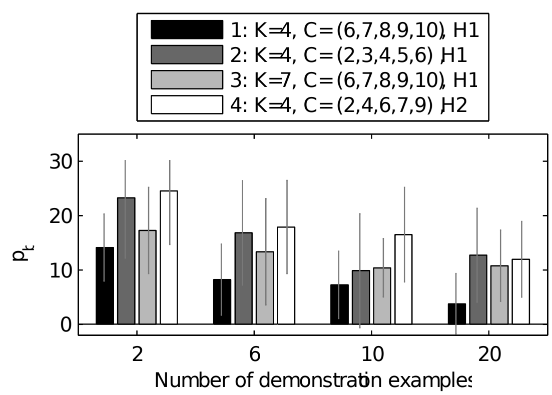

Figure 3. The percentage of times interaction style rules were broken for different experimental conditions, averaged over 20 trials. The error bars indicate the standard deviation of these percentages for each set of trials.

5. DISCUSSION AND CONCLUSION

Our aim was to adapt the approach of Omax and the Continuator, to high-level parameter control of an ensemble of virtual instruments. While we have not yet considered using a variable-length history as these systems do, the key feature of our IMA is that it does not require any presupplied knowledge of the parameters that it is controlling. We envisage its use as a high-level controller in a complex IMA that uses other machine learning techniques to manage lower level controls such as notes and gestures. The algorithm is easy to implement and compute, so it is suitable for use in parallel with other algorithmic techniques.

At the core of the proposed algorithm is the similarity measure for sequential patterns. This measure was chosen because, unlike other pattern similarity measures such as the *edit distance* [5], it takes into account both the contents of the pattern (i.e. the parameter values) and the order in which they occur. This means that immediate concerns–what combination of parameter values is appropriate here?–and structural concerns–is this part of a build-up of intensity, or a reduction?–can be satisfied with a good match found using the S^{2}MP.

Our current implementation is limited to binary-valued parameters. This can be easily extended to multi-valued parameters (see the use of dummy variables in [4]). In addition, the algorithm would be improved by the introduction of continuity rules to prevent the rapid changing of parameter values as was occasionally observed in real-time performance testing. These two features are currently being added.

6. APPENDIX

In this section, we give a mathematical description of the method by which the IMA chooses new parameter values. First, we introduce some notation. The number of sets in a set sequence $\mathscr{S}$ is denoted by $|\mathscr{S}|$. The symbol $\mathscr{S}_i$ is used to denote the ith set of $\mathscr{S}$, and $\mathscr{S}_{i:j}$ denotes the portion of $\mathscr{S}$ from the ith set to the jth set. Finally, we use $\{\mathscr{S}, A\}$ to denote the set sequence resulting from the addition of a set A to the end of a set sequence $\mathscr{S}$.

The performance set sequence (see Section 2) is denoted $\mathscr{P}$. Usually $\mathscr{P}$ contains K-sets, but if there have not yet been K-decision points in the performance, then $\mathscr{P}$ has as many sets as there have been decision points. The number of sets in $\mathscr{P}$ is denoted l.

As discussed in Section 2, the final set of $\mathscr{P}$ needs to be completed before the similarity between $\mathscr{P}$ and a set sequence, $\mathscr{Q}$, from the demonstration data set, can be calculated. This is done by modifying the last set of $\mathscr{P}$ so that it encodes the same IMA-controlled parameter values as the last set in $\mathscr{Q}$. We use $\mathscr{P}'$ to denote this modified version of $\mathscr{P}$. Formally, $\mathscr{P}' = \{\mathscr{P}_{1:l-1}, (\mathscr{P}_l \cup (C \cap \mathscr{Q}_l))\}$.

At a decision point, t_i, the IMA must choose values for the parameters in C. This choice can be encoded by a set of integers, C_1, corresponding to the parameters that will be set to 1. To choose new values, the IMA iterates over all length-l set sequences in the demonstration data set. For each of these, denoted $\mathscr{Q}$, it creates a corresponding $\mathscr{P}'$ and calculates the similarity score between $\mathscr{P}'$ and $\mathscr{Q}$. The parameter values chosen by the IMA are those encoded in the final set in the set sequence which gives rise to the highest similarity score. The procedure just described is listed in Algorithm 1, where the *sim* function (line 7) refers to the S^{2}MP algorithm.

Algorithm 1 The template-matching algorithm

1: $s_{max} \leftarrow 0$
2: $l \leftarrow |\mathscr{P}|$
3: **for** $n = 1$ to N **do**
4:     **for** $j = 1$ to $(|\mathfrak{D}^n| - l + 1)$ **do**
5:         $\mathscr{Q} \leftarrow \mathfrak{D}^n_{j:j+l-1}$
6:         $\mathscr{P}' \leftarrow \{\mathscr{P}_{1:l-1}, (\mathscr{P}_l \cup (C \cap \mathscr{Q}_{l+1}))\}$
7:         $s \leftarrow sim(\mathscr{P}', \mathscr{Q})$
8:         **if** $s > s_{max}$ **then**
9:             $s_{max} \leftarrow s$
10:             $C_1 \leftarrow (C \cap \mathscr{Q}_l)$
11:         **end if**
12:     **end for**
13: **end for**

7. REFERENCES

[1] G. Assayag and S. Dubnov, "Using factor oracles for machine improvisation," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 8, no. 9, pp. 604–610, 2004.

[2] J. Drummond, "Understanding interactive systems," *Organised Sound*, vol. 14, no. 2, pp. 124–133, 2009.

[3] M. Hamanaka, M. Goto, H. Asoh, and N. Otsu, "A learning-based jam session system that imitates a player's personality model," *Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico*, pp. 51–58, 2003.

[4] T. Hastie, R. Tibshirani, and J. H. Friedman, "The elements of statistical learning: data mining, inference, and prediction," *New York: Springer-Verlag*, 2009.

[5] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[6] G. Lewis, "Too many notes: Computers, complexity and culture in voyager," *Leonardo Music Journal*, vol. 10, pp. 33–39, Jan 2000.

[7] A. Martin, C. Jin, A. van Schaik, and W. L. Martens, "Partially observable markov decision processes for interactive music systems," *International Computer Music Conference, New York*, 2010.

[8] F. Pachet, "The continuator: Musical interaction with style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, Jan 2003.

[9] H. Saneifar, S. Bringay, A. Laurent, and M. Teisseire, "S2mp: Similarity measure for sequential patterns," *Proceedings of the Seventh Australasian Data Mining Conference*, 2008.