Chopin, mazurkas and Markov

Making music in style with statistics

How do people compose music? Can computers, with statistics, create a mazurka that cannot be distinguished from a Chopin original? **Tom Collins**, **Robin Laney**, **Alistair Willis** and **Paul H. Garthwaite** make music with Markov models.

Introduction

Long before the dawn of digital computing, random numbers were being used to generate passages of music. In a dice game supposedly invented by Wolfgang Amadeus Mozart (1756–1791), consecutive rolls of a pair of dice are used to select segments of pre-composed waltz-like music[1]. The segments are pieced together in the order they were selected and, after eight dice rolls, the player of the game has generated a passage of music that sounds like a waltz by Mozart. The claim that Mozart devised this game is probably false – a publisher's ruse to boost sales[2] – but the age of the game is evidence of an enduring interest in the automatic generation of music compositions in the style of a particular composer or period – stylistic compositions.

Computers of course have transformed the activity. More computer models for stylistic composition have focused on harmonising chorale melodies – adding alto, tenor and bass parts to hymn tunes – than on any other genre[3]. In particular, the chorales of Johann Sebastian Bach (1685–1750) underlie much work. Allan uses hidden Markov models to harmonise a chorale melody, treating the melody notes as observed states and determining, by empirical analysis of Bach's chorales, which hidden sequence of harmonies is most likely to underlie the given melody[4]. Ebcioğlu achieves similarly promising results via a completely different approach, encoding 350 musical "rules" and using logic programming to ensure that generated harmonisations obey the rules[5]. Rather than harmonising melodies, Pearce and Wiggins use existing hymn tunes as the basis for generating new melodies[6]. The only other classical musical genre that has attracted substantial attention is Baroque fugal exposition[7,8]. For contrast, we have focused on composing the opening section of a mazurka in the style of Frédéric Chopin (1810–1849). Mazurkas are Polish folk dances. Chopin used them as the inspiration for over 50 piano pieces, forming a good-sized corpus for research in stylistic composition.

Statistics can imitate artistic creativity. But can they do it well enough to deceive?

Copying out and reworking pieces by other composers is often the first step in a young composer's journey to "finding his or her own voice". The *cognitive process* by which the art of composition is learnt remains an enigma, however. If an algorithm can exploit existing pieces of music to generate successful stylistic compositions (as judged by independent and experienced listeners), then perhaps we can shed more light on the meaning of "musical style" and the human compositional process. Observing composers at work and asking them about the creative process are complementary methods for investigating the acquisition of compositional abilities, but the latter method in particular is susceptible to flights of fancy[9]. An algorithm for generating stylistic compositions might be adapted to assist students of music – it could offer students an initial fragment, or an appropriate continuation to a half-composed phrase. In England and Wales alone, an estimated 50,000 students each year respond to stylistic composition briefs in music exams. "Compose a short *Lied* in the style of Schumann" is a good example of a stylistic composition brief.

Markov meets "Three Blind Mice"

A Markov chain is a succession of states; each state depends only on the one that preceded it. A simple tune is a succession of notes. Assuming each note depends only on the note that preceded it, it can be analysed as a Markov chain.

Suppose in a tune that whenever a note of pitch C occurs, it is followed half of the time by a G, a quarter of the time by an E, and less frequently by other notes. Similar probabilities would apply to every other note in the octave. To take a real-life example, the tune "Three Blind Mice" can be written (ignoring octaves) as:

E D C, E D C,

G F F E, G F F E,

G C C B A B C G G,

G C C C B A B C G G,

G C C C B A B C G G G,

F E D C.

Illustration: Tom Boulton

The first note, E, occurs 5 times. Three of those times it is followed by D, twice it is followed by G, and it is never followed by any other note. D occurs 3 times. It is always followed by C.

A computer algorithm that generated a string of notes where D was always followed by C, and where E had a 3/5 chance of being followed by D and a 2/5 chance of being followed by G, would "compose" a "tune" that might be reminiscent of "Three Blind Mice".

Chopin is more complex, but can still be analysed, and imitated, by Markov chain algorithms.

"Three Blind Mice" also has near or exact repetitions of three- and four-note phrases, and longer ones as well. Sometimes these repetitions are at the same pitch, sometimes they are transposed higher. These patterns too can be incorporated in the composing algorithm – in nursery songs and in Chopin mazurkas.

Markov models of stylistic composition

The use of Markov chains is an important approach to algorithmic composition. The tune "Three Blind Mice" shows this very simply – see box. In that example, and below, pitch classes – that is, pitch regardless of octave – form the state space of the Markov chain, while the relative frequencies with which one pitch class leads to the following pitch class form the matrix of transition probabilities. (Other qualities of the notes, such as duration or timbre, could be used instead or as well.) We will illustrate this more fully using the melody in Figure 1. The piece of music contains all of the natural pitch classes as well as B♭, so the obvious choice for the state space (I) is the set of pitch classes

I = {F, G, A, B♭, B, C, D, E}.

The transition matrix in Table 1 records all the transitions between notes, with their relative frequencies. For example, there are four transitions from F, of which three are to G, while the fourth is to A. This gives the first row of the table: the transition probabilities are 3/4 from F to G, 1/4 from F to A, and 0 for other transitions. Each row of the table corresponds to transitions from a different pitch class. It can be seen that most transitions are from one pitch class to an adjacent one.

To use this matrix in a compositional scenario we start by choosing an initial note – say, A. We look along the A row of our table to choose our second note; we randomly choose between F, G, B and C, and with respective probabilities 1/8, 1/2, 1/4 and 1/8.

Suppose we choose B. Looking along the fifth row of Table 1, we select our third note, making a random, equiprobable choice between G, C, and D. And so on. We, or the computer, can use random (or pseudo-random) numbers to guide the choices at each note.

Every time we run the exercise, the resulting tune will be different . Below are three pitch sequences generated from the Markov model using pseudo-random numbers. For ease of reading, each melody is split up according to the phrase structure of the original music in Figure 1 (to hear all of these melodies, visit `http://www.tomcollinsresearch.net` and follow the links).

1. A, G, F, G, F, G, A, B, G,
   F, G, F, G, A, B, D, E,
   B, C, A, F, G,
   B♭, A, F, G, A, G, A, B, G, A.

2. A, G, A, B, D, C, B♭, A, F,
   G, F, A, B, D, C, A, G,
   A, G, F, A, F,
   A, F, G, F, G, A, G, F, A, G.

3. F, A, B, G, F, G, F, G, A,
   B, C, A, G, F, G, F, G,
   B♭, A, G, A, G,
   A, F, G, B♭, A, B, G, F, G, A.

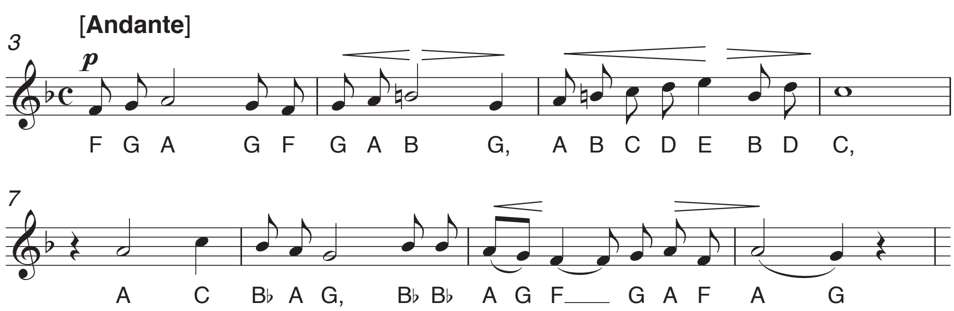

Figure 1. Bars 3–10 of the melody from "Lydia", Op. 4 No. 2, by Gabriel Fauré (1845–1924).

Table 1. Transition matrix for the material shown in Figure 1. The *i*th row and *j*th column records the number of transitions from the *i*th to the *j*th state in the melody, divided by the total number of transitions from the *i*th state.

Pitch class	*F*	*G*	*A*	*B♭*	*B*	*C*	*D*	*E*
F	0	3/4	1/4	0	0	0	0	0
G	2/7	0	4/7	1/7	0	0	0	0
A	1/8	1/2	0	0	1/4	1/8	0	0
B♭	0	0	2/3	1/3	0	0	0	0
B	0	1/3	0	0	0	1/3	1/3	0
C	0	0	1/3	1/3	0	0	1/3	0
D	0	0	0	0	0	1/2	0	1/2
E	0	0	0	0	1	0	0	0

The above example of constructing a Markov model and using it to generate pitch sequences raises several questions. First, the majority of classical music is polyphonic (more than one pitch is sung/played simultaneously), but above we modelled a monophonic excerpt (only one pitch is sung/played at a time). How should the definition of "state" be altered to build analogous Markov models for polyphonic music? Second, when a transition matrix is constructed using one or more pieces of music, how can we prevent generated passages replicating substantial parts of existing work? Third, repeated patterns play an important role in music, so how can we ensure that a generated passage contains repeated patterns, be they short motifs or longer sections?

In answer to the first question, one plausible definition of a polyphonic state is a *set* of pitches, as opposed to *lone* pitches. For example, the set {F, A, C} is a state that might be followed by the state {E, G, B♭, C}. Another plausible definition involves counting the interval in semitones between simultaneous pitches, when arranged in ascending order. For example, there are four semitones from F to A, and three semitones from A to C, so (4, 3) is a state in such a state space. Determining the best choice of state space for polyphonic music is an open problem.

Turning to the second question – how to avoid replicating too much of the original composer – it is possible to retain the source information (e.g., Fauré, Op. 4 No. 2) for each observed state. In this way, we can impose a constraint on the generation process, stipulating that no more than four consecutive generated states, say, may have the same source. This constraint reduces the likelihood of the generated passage replicating substantial parts of existing work.

To address the third question, on repetitions, we introduce another topic where music and statistics intersect: algorithms for discovering repeated patterns in music.

Pattern discovery and pattern inheritance

It is uncontroversial that *repetition* plays a central role in our perception of musical structure: "Only by repetition can a series of tones be characterized as something definite. Only repetition can demarcate a series of tones and its purpose. Repetition thus is the basis of music as an art"[10]. Hence, pattern discovery and pattern inheritance should play a central role in the algorithmic generation of music.

A pattern discovery algorithm takes a symbolic representation of a single piece of music as input, and produces a list (or graphical representation) of the repeated patterns that occur in the piece. Bioinformatics algorithms that were originally intended for discovering repetitions in DNA *strings* are easily adapted to monophonic music, as such music can be represented as strings of pitches and/or durations[11]. However, another approach, which works as well for polyphony as it does for monophony, is to use a *point-set* (or *geometric*) representation of a piece in order to discover repeated patterns[12]. Methods that use this approach have been developed by Meredith, Lemström and Wiggins, and we have extended one of their methods to give an algorithm we call SIACT (Structure Induction Algorithm with Compactness Trawling)[13]. Output from this algorithm is illustrated in Figure 2, where it was applied to an excerpt from a Chopin mazurka.

Musical patterns can be shifted as a block by a number of notes (either up or down) and our minds will hear the repetition. Analytically,

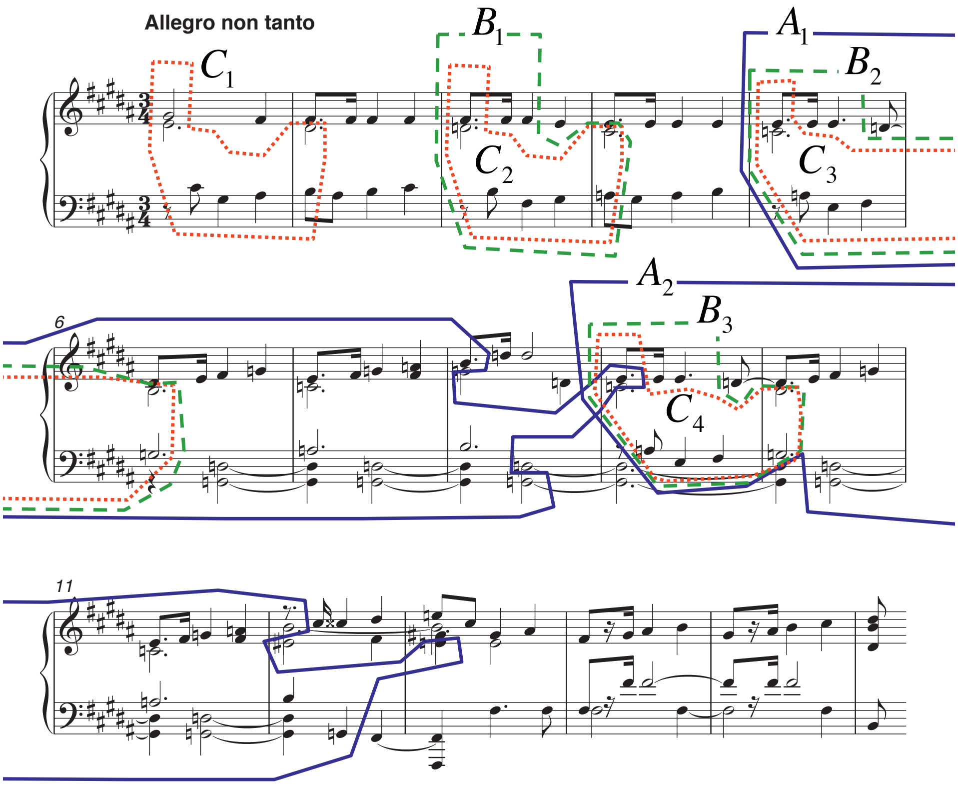

Figure 2. SIACT was applied to a representation of bars 1–16 of the Mazurka in B major, Op. 56 No. 1, by Chopin, and the results were filtered and rated. Occurrences of the top three patterns are shown.

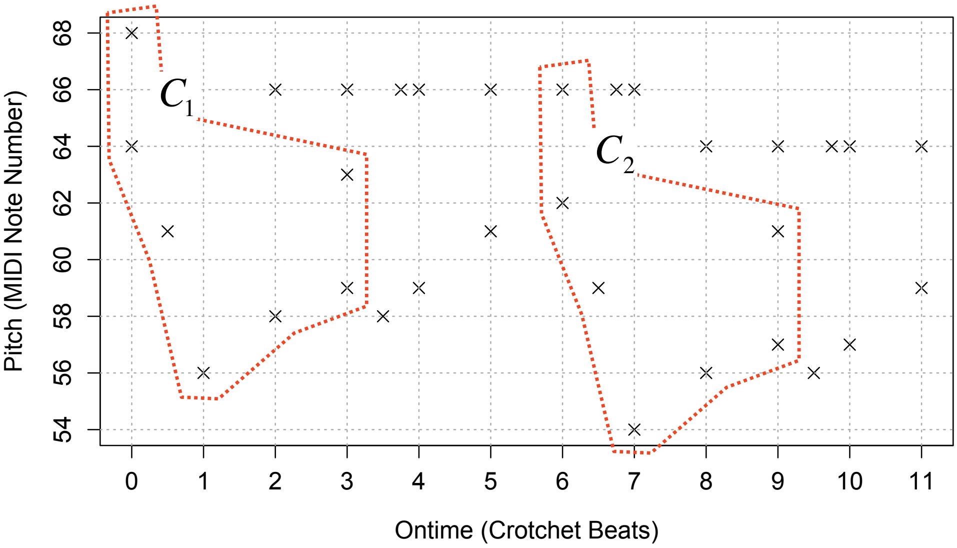

Figure 3. Point-set representation of the beginning of Figure 2. The subset of points labelled C_1 is called a translational pattern, as translating each element of C_1 by the vector (6, −2) gives the subset of points labelled C_2.

it is possible to discern this type of repetition by discovering corresponding translational subsets in geometric representations. For instance, Figure 3 contains a point-set representation of the first few bars of music given in Figure 2. The subset of points labelled C_1 is repeated six beats later and two semitones lower, as C_2. It is called a translational pattern and the vector (6, −2) describes how C_1 leads to C_2. These point-sets correspond to the note collections annotated as C_1 and C_2 in Figure 2. Beyond (or abstracted from) the domain of music, the problem of defining algorithms for discovery of translational point-sets seems to us a worthy topic of research in its own right.

In all but the shortest excerpts of music, it is likely there will be a large number of repeated patterns, so it is useful to rate and perhaps discard some of the output of a pattern discovery algorithm, keeping only the most musically important patterns. Conklin and Bergeron adapt statistical methods such as the likelihood ratio, renaming it *pattern interest*, to try to differentiate between patterns that are perceived as musically important and those that are not[11]. We have used variable selection and cross-validation to build models for predicting the perceived salience of a discovered pattern, using novel quantifiable pattern attributes, as well as attributes defined by Conklin and Bergeron[11], Meredith *et al.*[12], Pearce and Wiggins[6], and others. The participants in one of our studies were 12 Cambridge University music students[14]. Their ratings of a musical pattern's importance could be modelled by three factors: compactness, a compression ratio, and the expected number of occurrences of the pattern. During the time interval spanned by a pattern, some notes may be played that are not part of the pattern, and *compactness* measures the proportion of contemporaneous notes that are in the pattern. The *compression ratio* is (approximately) the harmonic mean of the number of notes in a pattern and the pattern's number of occurrences. High values for compactness and the compression ratio increased a pattern's perceived importance, as one might anticipate, while being unexpected also made a pattern more important.

Once important repeated patterns have been discovered in an existing piece of music, how can these discoveries be used to guide the generation of a new passage? An approach stemming from Cope's work[8] is to retain the temporal and registral positions of the repeated patterns – not the actual notes – in a so-called *pattern template*. The template for Figure 2 is shown in Figure 4, and can be used to guide the music generation process as follows. Material for the red box labelled C_3 will be generated first, as this is the *most nested* pattern. Second, copies of the material generated for C_3 will be translated appropriately, filling the boxes labelled C_1, C_2, and C_4. Third, material for the blue box labelled A_1 will be generated. (The next most nested pattern is actually B_2, not A_1, but the former is overlooked because, according to the template in Figure 4, it is no different from C_3, which has already been filled.) The blue box A_1 is already

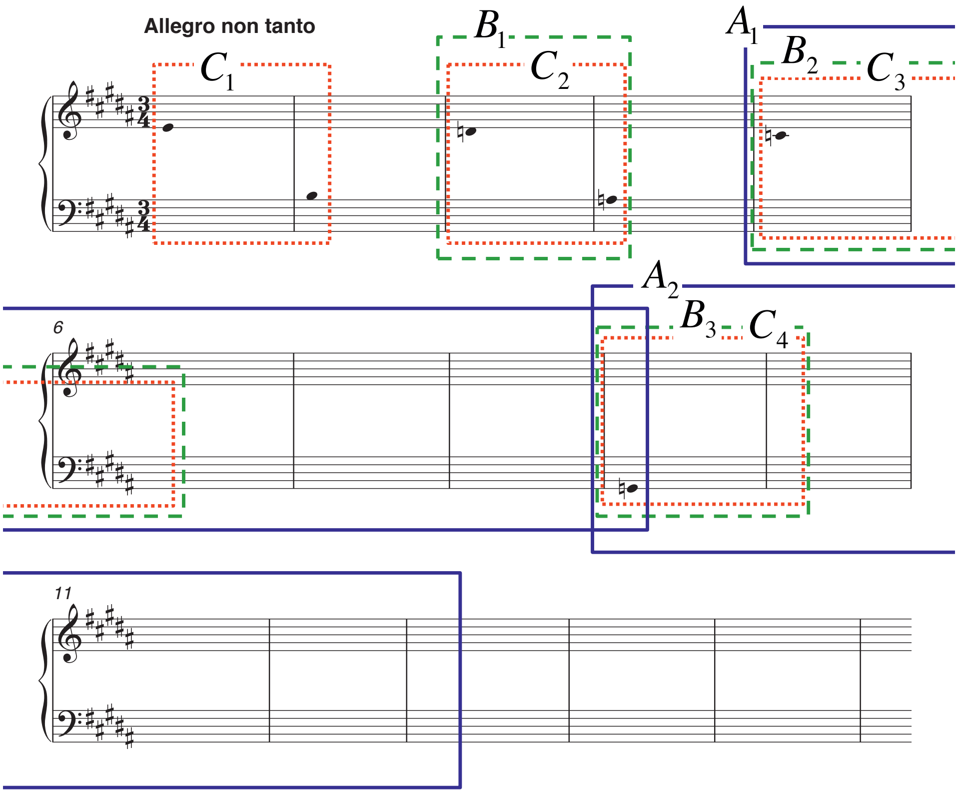

Figure 4. Representation of a pattern template. Most of the content of the excerpt from Figure 2 has been removed, but the temporal and registral positions of the discovered patterns remain.

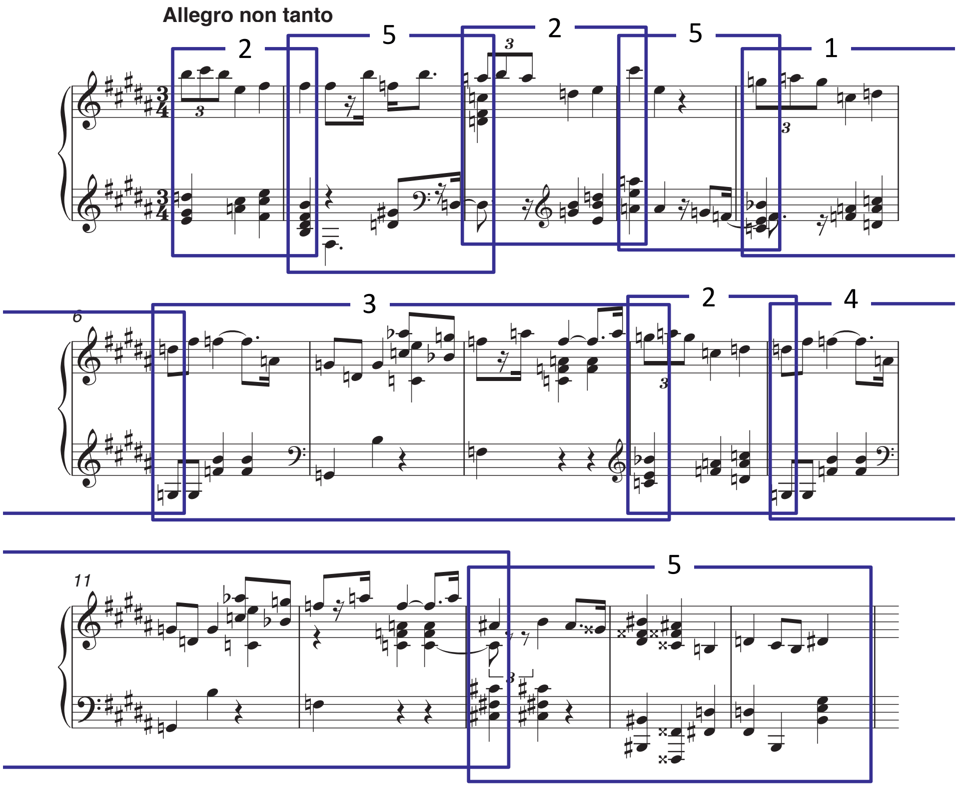

Figure 5. Passage generated by the Racchmaninof-Oct2010 model. The numbered boxes indicate the order in which different parts of the passage are generated, and correspond to an enumerated discussion in the prose.

part filled with material from C_3, but the remainder is filled, and in the fourth step a copy is translated appropriately, filling box A_2. The fifth and final step is to fill all of the remaining empty bars of the template. An example of the output from this generation process is shown in Figure 5, where the steps are labelled. It inherits the repetitive structure – not the actual notes – from Figure 2. We call this *pattern inheritance*. Our model for generating these passages is called Racchmaninof-Oct2010, which with no contrivance at all stands for RAndom Constrained CHain of MArkovian Nodes with INheritance Of Form, with a date stamp (for the source code, visit `http://www.tomcollinsresearch.net` and follow the links). The collection of databases and programs referred to by Cope as Experiments in Musical Intelligence (EMI) appears to make use of pattern inheritance as well, although this is not a topic that Cope discusses at length[8].

Is the generated music any good?

In an experiment we conducted[15], 32 participants (16 recruited from a concert audience and 16 postgraduates with an interest in nineteenth-century music) were asked to listen to and answer questions about short excerpts of music. Some of the excerpts were genuine Chopin mazurkas, some were human-composed but not Chopin mazurkas, some were from the output of Cope and EMI, and some were from the output of Racchmaninof-Oct2010. Participants rated the stylistic success of each excerpt, relative to their understanding of a typical Chopin mazurka, and also gave a rating for aesthetic pleasure. In addition, they had to say whether they thought the music was a Chopin mazurka, human-composed but not by Chopin, or generated by a computer algorithm. Written comments on the basis of their judgements were also solicited. The format was similar to the consensual assessment technique used by Amabile to determine the perceived creativity of a set of artistic products[16]. Another evaluation format that springs to mind in this context is the Turing test, where a human participant engages in blind questioning of (1) a computer program and (2) a female, and must decide on the basis of the answers, which is actually the female[17].

Analysis of the judges' responses in our study suggests that some aspects of musical style are being modelled effectively by Racchmaninof-Oct2010. All but one of the excerpts from Racchmaninoff-Oct2012 (as in Figure 5) were rated by the postgraduate judges as more stylistically successful than a mazurka by an amateur composer, and sometimes passages generated by our model were difficult to distinguish from original Chopin mazurkas. That said, the results also indicate potential for future improvements, particularly with regard to aspects of harmony.

Do the judges' comments shed any light on listening strategies for distinguishing between human-composed and computer-generated music? It can be difficult to articulate the reasoning that leads to deciding one way or the other, and perhaps this is reflected by similar comments from judges that lead to different decisions: one excerpt from a real Chopin mazurka was miscategorised by a concertgoer judge, with an observation that "the intro seemed not in character"; whereas a postgraduate judge categorised it correctly, observing that it is "harmonically…complex but also goes where one hopes it will. Slightly unusual opening (solo right hand), but seems to get going after this". Similarly, judges were sensitive to random-sounding aspects of excerpts, but vacillated over whether or not *randomness* indicated a computer-based source or a human one. In relation to the same excerpt, for instance, a concertgoer judge observed "it sounds too random to be computer generated", whereas for another concertgoer judge the "rhythm was mostly OK but the random melodic line seems computerish". Finally, although this may not have had a bearing on the distinguishing question, postgraduate judges appeared to be more receptive than concertgoer judges to an atonal excerpt by Arnold Schoenberg (1874–1951): "love it – it sounds almost 12-tone" (a postgraduate judge); "could well be by a modern composer, not my cup of tea, a computer program would do better than this" (a concertgoer judge).

Implications for human musical creativity

Looking to the future, will an algorithm such as ours, using existing music databases, ever be capable of generating compositions that are judged as stylistically successful as pieces from the intended style? Our answer is a tentative "sometimes, though an algorithm's success rate would depend on the style to be emulated". Will such a capability inhibit or destroy human musical creativity? Almost certainly not, although we foresee students of composition being able to dip into diverse styles of music by requesting novel initial fragments or appropriate continuations from Markov-based algorithmic assistants. Moreover, the capability of generating stylistically successful compositions has the potential to enhance the way in which humans create music, in much the same way that existing music software, such as GarageBand, Songsmith, and Hyperscore, enables non-experts to compose. Research has only begun to scratch the surface of how humans learn the art of composition and the role that computer algorithms could play.

Painting by Amy Collins (www.amycollinspaintings.co.uk). Mixed media on board

References

1. IMSLP (2010) Musikalisches Würfelspiel. http://imslp.org/wiki/Musikalisches_Würfelspiel,_K.516f_(Mozart,_Wolfgang_Amadeus)

2. Hedges, S. A. (1978) Dice music in the eighteenth century. *Music and Letters*, **59**, 180–187.

3. Nierhaus, G. (2009) *Algorithmic Composition: Paradigms of Automated Music Generation*. Vienna: Springer.

4. Allan, M. (2002) Harmonising chorales in the style of Johann Sebastian Bach. Master's thesis, School of Informatics, University of Edinburgh.

5. Ebcioğlu, K. (1994) An expert system for harmonizing chorales in the style of J. S. Bach. In *Understanding Music with AI: Perspectives on Music Cognition* (eds. M. Balaban, K. Ebcioğlu, and O. Laske), pp. 145–185. Menlo Park, CA: AAAI Press.

6. Pearce, M. T. and Wiggins, G. A. (2007) Evaluating cognitive models of musical composition. In *Proceedings of the International Joint Workshop on Computational Creativity*, pp. 73–80. London: Goldsmiths, University of London.

7. Craft, A. and Cross, I. (2003) A n-gram approach to fugal exposition composition. In *Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pp. 36–41. Brighton: SSAISB.

8. Cope, D. (2005) *Computer Models of Musical Creativity*. Cambridge, MA: MIT Press.

9. Collins, D. (2005) A synthesis process model of creative thinking in music composition. *Psychology of Music*, **33**, 193–216.

10. Schenker, H. (1973) *Harmony* (transl. E. Mann Borgese, ed. O. Jones). Cambridge, MA: MIT Press. (Original work published in 1906 by Cotta, Stuttgart.)

11. Conklin, D. and Bergeron, M. (2008) Feature set patterns in music. *Computer Music Journal*, **32**, 60–70.

12. Meredith, D., Lemström, K. and Wiggins, G. A. (2002) Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, **31**, 321–345.

13. Collins, T., Thurlow, J., Laney, R., Willis, A. and Garthwaite, P. H. (2010) A comparative evaluation of algorithms for discovering translational patterns in Baroque keyboard works. In *Proceedings of the International Symposium on Music Information Retrieval*, pp. 3–8. Utrecht: International Society for Music Information Retrieval.

14. Collins, T., Laney, R., Willis, A. and Garthwaite, P. H. (2011) Modelling pattern importance in Chopin's mazurkas. *Music Perception*, **28**, 387–414.

15. Collins, T., Laney, R., Willis, A. and Garthwaite, P. H. (forthcoming) Developing and evaluating computational models of musical style. Submitted for publication.

16. Amabile, T. M. (1996) *Creativity in Context*. Boulder, CO: Westview Press.

Tom Collins is a postdoctoral researcher at the Center for Mind and Brain, University of California, Davis. His first degree was in music, and he also has degrees in mathematics and statistics, and in computing. Robin Laney and Alistair Willis are a Senior Lecturer and Lecturer in the Department of Computing and Paul Garthwaite is a professor in the Department of Mathematics and Statistics, all at the Open University.