Examiners' commentary 2018–2019

CO3346 Creative computing: sound and music – Zone B

General remarks

Overall performance on this paper was excellent, with no fails and one candidate obtaining almost full marks.

What follows is a brief discussion of the individual questions on this paper, with hints towards the answers expected by the examiners.

Comments on specific questions

Question 1: Computational models of music cognition

Performance on this question was mostly good, with candidates showing both knowledge of the basic information and also understanding of concepts.

a. For part (a), it was important to include information about computational modelling as well as a description of what cognitive science is. A good answer might say that cognitive science is the study of the way people make sense of, and interpret information, while computational models are used to embody theories of cognition in order to evaluate those theories.

b. Part (b) required candidates to identify and describe two application areas, and they needed to relate to musical cognition. Examples might include: algorithmic composition, design of compositional tools, computational music analysis or music cognition. Marks were awarded for correctly naming, as well as providing an adequate description of the areas chosen.

c. The main requirement for part (c) was to provide clear descriptions of the characteristics of both kinds of representations, and then example file formats that are used to store representations of that kind. For a sequence of notes, for example, the characteristics would likely be things such as it being symbolic, with pitch, duration and time-stamps being represented. An appropriate format would be a MIDI file, or a text file containing data.

d. For part (d), many candidates did not seem to understand what was being asked and therefore gave somewhat random responses. Examples of appropriate analyses might be to identify different sections in a piece of music or to identify different instrumental parts.

e. In responding to part (e), most candidates realised that this was related to the Gestalt principles studied in the course. As an example of specific principles, one might be the principle of proximity, and it could be specifically related to time, where notes near to each other might be in the same melody. Other reasonable suggestions were also accepted, provided they were properly described.

f. For part (f), any reasonable examples were acceptable, as long as they were musical models. However, it was essential to clarify what the suggested model would be able to predict. For example, a two-factor model of melodic expectation predicts which note a person is most likely to expect after listening to a note sequence.

g. Part (g) was answered well by some candidates, though not all included examples that demonstrated full understanding. To build an n-gram model, you would divide the note sequence into sets of n notes, with the next note that happens after each sequence. For example, if n is 2, you might have [a, b flat] goes to c sharp.

h. Most candidates suggested the Krumhansl-Schmuckler key-finding algorithm for part (h), though other algorithms do exist and would have been acceptable. For Krumhansl-Schmuckler, the input is a set of notes with pitches and duration, and the output is a key. It does not use an n-gram model as it does not split the piece into sub-sequences.

Question 2: Computer music and pure data

This was a fairly popular question, answered reasonably by some of the candidates who chose it. The main weakness was that candidates failed to provide adequate responses to all parts of the questions, thereby not obtaining the full number of marks available.

a. For part (a), examiners expected candidates to identify sensible differences, and then for each, ways in which Pure Data and a text language differ in terms of their IDE. A sensible difference might be, for example, compiler output. The Java netbeans IDE outputs the result of compilation. Pure Data does not compile so this is not needed. A complete answer would cover three differences in this way.

b. Similarly, for part (b), examiners expected at least five steps in the instructions for how to build a patch, as well as diagrams. Steps should have included the loading for the file, as well as the play back by the patch.

c. To obtain the two marks for part (c), candidates were expected to explain that synthesized sound is sound that has been generated from periodic or noise wave-forms. Candidates were also asked to describe how synthesized sound is different from sampled sound.

d. Finally, for part (d) many candidates were able to draw a correct Pure Data patch. However, not all candidates supplied the corresponding annotation, and therefore failed to achieve the full four marks available.

Question 3: Algorithmic composition

This was a very popular question, chosen by almost all candidates and answered reasonably well by most.

a. For part (a), most candidates knew that the example was algorithmic, but not all explained clearly why this is the case. The composer is not composing the notes themselves, they are using the dice to compose the notes.

b. Part (b) was a straightforward question requiring candidates to name a composer who has used algorithmic composition. The most obvious response would be Xenakis. However, any other correct answers were accepted, though a couple of names offered were not actually algorithmic composers. For example, Devin Townsend, who is a rock musician but not an algorithmic composer. Some candidates correctly identified Mozart.

c. Many candidates chose the example of swarm music for their answer to part (c). Another appropriate choice could be the technique of pulling patterns from nature, such as Xenakis' rain drop percussion. In all cases, a description was also required to obtain full marks.

d. Part (d) was answered correctly by almost all candidates, who were able to say that this is live coding.

e. For part (e), some interesting reasons for live coding were given. For example, the fact that it is interesting for the audience to see the algorithm being written, and also the fact that this is technically challenging to do and therefore virtuosic.

f. Part (f) was straightforward material from the subject guide requiring candidates to name and describe four desirable characteristics of live algorithms. The desirable characteristics are reflection, innovation, transparency and autonomy, which most candidates were able to list. Some candidates failed to provide clear descriptions, and therefore did not gain the full marks available for this question.

g. For part (g), candidates were required to apply the same four concepts identified in part (f) to the scenario of jazz improvisation, again explaining or justifying the choices made.

h. Part (h) asked candidates to explain the key elements of a live algorithm based on a swarm model. The key elements to explain would be: the particles that make up the swarm, a visualisation, a mapping from swarm to music and the swarm movement algorithm or algorithms.

Question 4: Music information retrieval

This question was chosen by many candidates, and generally answered reasonably well, though some candidates failed to answer the final part.

a. Part (a) asked candidates to name two music information retrieval tasks, one that has high specificity and one that has low specificity. The examiners expected clear examples, for example music identification has high specificity while music-speech segmentation has low specificity.

b. Part (b) was generally answered weakly, as many candidates did not include all aspects. It was best to choose a particular recommender system as an example, such as the Genius feature in Apples iTunes music player, which establishes commonalities between tracks on a user's system with iTunes playlists containing those tracks created by other iTunes users. The commercial motivation is clear, the recommendations generated by the Genius feature come with the ability to purchase recommendations corresponding to tracks not already in the user's collection.

   In terms of how they work, recommender systems such as this use a large, previously acquired set of data, whether from having access to all playlists, purchase data or prior musical feature extraction and analysis. They will therefore only be able to recommend tracks (or however musical works are represented) that exist in their own databases. This is not so useful for exploring niche, targeted collections (or in the case of Apple's iTunes, for example, recommendations based on the Beatles collection).

   In contrast, there are recommender systems based on navigating a standalone collection, where the emphasis is not on finding new material to offer a user, but on providing a path to navigate through the existing material. This is a necessary tool in this age of consumer electronic devices with tens of thousands of tracks on them. Some candidates included appropriate diagrams, which also attracted marks.

c. For part (c), it was possible to either agree or disagree with the appropriateness of timbre as a feature for building a recommender system. Whichever candidates chose, it was essential for them to justify their answer. For example, the system might find tracks with similar instruments or similar voices, and a discussion of how this relates to recommendation would also be important.

d. Part (d) required the understanding that data can be the content itself, or the metadata. Explaining what these are or giving examples enhanced the responses and saw candidates gain the full two marks available for this question.

e. For part (e), relating this to part (d) was essential, as was a justification of which choice was made. Metadata would be the most appropriate. For example, if you know the artist's name, you can assign the most common genre for all of their other music. Content based would be less reliable as not all music in a genre actually sounds the same.

f. Finally for part (f), candidates were expected to consider how a recommender system would work, the kinds of data it would need, and the sorts of output it might provide. Based on this, for each of the three example scenarios (i–iii), they were then expected to describe how the system would provide suggestions. Many, but not all, candidates provided a coherent answer for this. Some answered for each of the three in different ways, implying that three different systems were being proposed, which was not what was expected. As an example of what was expected, for sub-part (iii), if there is no information about the user, it might be appropriate to select the current most popular tracks on the network. Data required might be their age and where they live.