Music Analysis and Data Compression

David Meredith

Department of Architecture, Design and Media Technology

Aalborg University

dave@create.aau.dk

Author's version of Chapter 47 of

The Oxford Handbook of Sound and Imagination, edited by Mark Grimshaw, Mads Walther-Hansen and Martin Knakkergaard (OUP, 2018)

1 Introduction

Most people are capable of imagining music, and composers can even imagine novel music they have never heard before. This is known as *musical imagery* and can be distinguished from *musical listening* or *music perception*, where the music experienced results from physical sound energy being transmitted across the listener's peripheral auditory system and then transduced in the inner ear into nerve signals that are propagated to higher centers of the brain. In both music perception and musical imagery, what is experienced is actually an *encoding* of musical *information*, created by the person's brain. Alternatively, one could adopt a less dualist stance and say that experiencing music is the direct result of certain spatio-temporal patterns of neural firing that encode musical information. In listening, this encoding is generated from information about sound currently in the environment, combined with the person's musical knowledge. In imagery, the encoding is constructed *only* from the person's musical knowledge. Sound is thus just one particular medium for communicating musical information and is not a prerequisite for musical experience. Indeed, trained musicians can experience (i.e., "imagine") music they have never previously heard while

silently reading a musical score. Musical imagery and perception therefore have a great deal in common – indeed, there are some brain centers (especially in the right temporal lobe) that are necessary for both (Halpern 2003).

Both the way that one perceives and understands music as well as the music that one is capable of imagining are therefore largely determined by one's musical knowledge that is gained through passive exposure to music, active learning of musical skills, and/or study of music theory and analysis. It has been proposed in psychology, information theory, and computer science that knowledge acquisition – that is, *learning* – is essentially *data compression* (Chater 1996; Vitányi and Li 2000): on being exposed to new data, a learning system attempts to encode this data as parsimoniously as possible by removing redundancy in the data and relating it to what it already knows. If a learning system can describe the new data in a compressed manner, then the total amount of space used to store all the system's knowledge increases by only a small amount. The less extra space required to encode the new data, the better this new data is "understood" by the system.

In this chapter, I focus on how the musical knowledge that underpins both music perception and musical imagery can be acquired by compressing musical information. In particular, my concern is with how it might be possible to find the *best* ways of understanding musical works simply by compressing as much as possible the information that they contain. The ideas presented in this chapter are founded on the assumption that the goal of music analysis is to find the best possible explanations for the structures of musical "objects," where such objects are typically individual works or movements but could be extracts from works (e.g., phrases, chords, voices, even individual notes) or collections of works (e.g., all the pieces by a composer or in a particular genre). This assumption begs the following question: given two analyses of the same musical object (i.e., two different explanations for the object or ways of understanding it), how are we supposed to decide which of the two is the "better" one?

Musicologists and music analysts who adopt a humanistic approach generally do not use objective criteria for deciding which of two possible analyses of the same musical object is preferable. Typically, such scholars prefer analyses that provide what they individually consider to be "more satisfying" readings of a work – that is, the ones that make them *feel* as though they have a better understanding of a work or repertoire. Analysts, then, traditionally evaluate musical analyses on subjective grounds. However, claiming that one analysis of a piece of music is "better than" another one for the same piece is considered here to be meaningless, unless one specifies *objectively evaluable tasks* that the first analysis allows one to perform more successfully. If such tasks are specified, then one can then meaningfully aim to find those analyses that are the best for carrying out those tasks. Such tasks could include:

- memorizing a piece in order, for example, to be able to perform it without a score;
- identifying errors in a score or performance of an analyzed piece or other related pieces;
- correctly identifying the composer, place of composition, genre, form, and so on of an analyzed piece or other related pieces;
- predicting what occurs in one part of a piece, having analyzed another part of the same piece or other related pieces; or
- transcribing a performance of a piece from an audio recording or MIDI representation to staff notation.

Of course, it may be the case that there is no *single* way of understanding a piece or set of pieces that allows for optimal performance on *all* such tasks. For example, the best way of understanding a piece in order to be able to detect errors in a performance may not be the best way of understanding that piece in order to determine if some other, previously unheard, piece is by the same composer. There may also be several different ways of understanding a given piece or set of pieces that are equally effective for carrying out a given task. Nevertheless, it will often be the case that understanding a piece in certain ways will allow one to carry out certain objectively evaluable

tasks more effectively than understanding the piece in certain other ways; to this extent, one can speak of some analyses as being "better than" others *for carrying out specific, objectively evaluable tasks.* The goal of the work presented in this chapter is therefore that of finding those ways of understanding musical objects that allow us to most effectively carry out the musical tasks that we want to accomplish. The approach adopted is based on the hypothesis that the best possible explanations for the structure of a given musical object are those that

1. are as simple as possible;
2. account for as much of the detailed structure of the object as possible; and
3. set the object in as broad a context as possible (i.e., describe the object as being part of as large a body of music as possible).

Clearly, these goals often conflict: accounting for the structure of a piece in more detail or in a way that relates the piece to all the music in some larger context can often entail making one's explanation (i.e., analysis) more complex.

This hypothesis, which forms the foundation for the work reported in this chapter, is a form of the well-known principle of parsimony. This principle can be traced back to antiquity[1] and is known in common parlance as "Ockham's razor," after the medieval English philosopher, William of Ockham (ca. 1287–1347), who made several statements to the effect that, when presented with two or more possible explanations that account for some set of observations, one should prefer the simplest of these explanations.

In more recent times, the parsimony principle has been formalized in various ways, including Rissanen's (1978) *minimum description length* (MDL) principle, Solomonoff's (1964a; 1964b) theory of inductive inference and Kolmogorov's concept of a minimal algorithmic sufficient statistic[2] (Li and Vitányi 2008, 401ff; Vereshchagin and Vitányi 2004). The essential idea underpinning these concepts is that explanations for data (i.e., ways of understanding it) can be derived from it by *compressing* it – that is, by finding parsimonious ways of describing the data

by exploiting regularity in it and removing redundancy from it. Indeed, Vitányi and Li (2000, 446) have shown that "data compression is almost always the best strategy" both for model selection and prediction.

The basic hypothesis that drives the research presented in this chapter is thus that the more parsimoniously one can describe an object without losing information about it, the better one *explains* the object being described, suggesting the possibility of automatically deriving explanatory descriptions of objects (in our case, musical objects) simply by the lossless compression of 'in extenso' descriptions of them. In the case of music, such an in extenso description might, for example, be a list of the properties of the notes in a piece (e.g., the pitch, onset, and duration of each note), such as can be found in a MIDI file. Alternatively, it could be a list of sample values describing the audio signal of a musical performance, such as can be found in a PCM audio file. The defining characteristic of an in extenso description of an object is that it explicitly specifies the properties of each *atomic component* of the object (e.g., a MIDI event in a MIDI file or an audio sample in a PCM audio file), without grouping these atoms together into larger constituents and without specifying any structural relationships between components of the object.[3] In contrast, an *explanation* for the structure of an object, such as an analysis of a musical object, will group atomic components together into larger constituents (e.g., notes grouped into phrases and chords or audio samples grouped together into musical events), specify structural relationships between components (e.g., "theme *B* is an inversion of theme *A*"), and classify constituents into categories (e.g., "chords *X* and *Y* are tonic chords in root position," "bars 1–4 and 16–19 are occurrences of the same theme"). Throughout this chapter, I assume that an analysis is a *losslessly* compressed encoding of an in extenso description of a musical object, even though most musical analyses to date have typically been *lossy*, in that they only focus on certain aspects of the structure of an object (e.g., harmony, voice-leading, thematic structure, etc.). Such *lossily* compressed encodings of an object can also provide useful ways of understanding it, but, because

information in the original object is lost in such encodings, they do not (at least individually) explain all of the detailed structure of the object. In particular, such lossy encodings do not provide enough information for the original object to be exactly reconstructed. Thus, if one is interested, for example, in learning enough about a corpus of pieces in order to compose new pieces of the same type, then such lossy analytical methods would not be sufficient.

In the remainder of this chapter, it is proposed that a musical analysis can fruitfully be conceived of as being an algorithm (possibly implemented as a computer program) that, when executed, outputs an in extenso description of the musical object being analyzed, and thus serves as a hypothesis about the nature of the process that gave rise to that musical object. Moreover, it is hypothesized that, if one has two algorithms or programs that each generate the same musical object, then the *shorter* of these (i.e., the one that can be encoded using fewer bits of information) will represent the *better* way of understanding that object for any task that requires or benefits from musical understanding.

A model of music perception and learning will be sketched later on in this chapter, that is based on the idea of accounting for the structure of a newly experienced piece of music by minimally modifying a compressed encoding of previously encountered pieces. Some recent work will then be reviewed in which these ideas have been put into practice by devising compression algorithms that acquire musical knowledge that can then be applied in automatically carrying out a variety of advanced musicological tasks.

2 Encodings, decoders, and two-part codes

In this chapter, a *musical analysis* is conceived of as an effective procedure (i.e., algorithm), possibly implemented as a working computer program, that, when executed, generates as its only output an *in extenso* description of the music to be explained. Typically, the description of this

program may be *shorter* than its output. A basic claim of this chapter is that such a description (in the form of a program) becomes an *explanation* for the structure of the object being described as soon as it is *shorter* than the in extenso description of the object that it generates. In other words, a compressed encoding of an in extenso description of an object can be considered a candidate *explanation* (not necessarily a "correct" one) for the structure of that object because it serves as a hypothesis as to the nature of the process that gave rise to the object.

Moreover, it is hypothesized that the more parsimoniously one can describe an object on some given level of detail, the better that description explains the structure of the object on that level of detail. As discussed above, this is an application of Ockham's razor or the minimum description length (MDL) principle (Rissanen 1978).

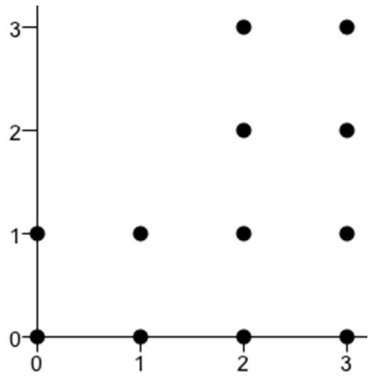

FIGURE 1. A set of 12, two-dimensional points in a Euclidean integer lattice.

The following simple example serves to illustrate the foregoing ideas. Consider the problem of describing the set of 12 points shown in figure 1. One could do this by explicitly giving the co-ordinates of all 12 points, thus:

P(p(0,0),p(0,1),p(1,0),p(1,1),p(2,0),p(2,1),p(2,2),p(2,3),p(3,0),p(3,1),p(3,2),p(3,3)) . (1)

In this encoding, a set of points, $\{p_1, p_2, \ldots p_n\}$, is denoted by P(p_1, p_2,… p_n) and each point within such a set is denoted by p(x,y), where x and y are the x- and y-co-ordinates of the point

respectively. The encoding in (1) can be thought of as being a program that computes the set of points in figure 1 simply by specifying each point individually. Representing this set of points in this way requires one to write down 24 integer co-ordinate values. Moreover, the encoding does not represent any groupings of the points into larger constituents, nor does it represent any structural relationships between the points. In other words, this description is an in extenso description that does not represent any of the structure in the point set and therefore cannot be said to offer any *explanation* for it. One could go even further and say that expression (1) represents the data as though it were a *random*, *meaningless* arrangement of points with no order or regularity.

Note that, in order to actually generate the set of 12 points, the description (1) needs to be *decoded*. An algorithm that carries out this decoding is called a *decoder*. In this case, such a decoder only needs to know about the meanings of the P(·) and p(*x*,*y*) formalisms.

One can obtain a shorter encoding of the point set in figure 1 by exploiting the fact that it consists of three copies, at different spatial positions, of the square configuration of points,

P(p(0,0),p(0,1),p(1,0),p(1,1)) . (2)

One could represent this description of the point set as follows:

T(P(p(0,0),p(0,1),p(1,0),p(1,1)),V(v(2,0),v(2,2))) , (3)

where T(P(p_1, p_2,… p_n),V(v_1, v_2,… v_m)) denotes the union of the point set, {p_1, p_2,… p_n}, and the point sets that result by translating {p_1, p_2,… p_n} by the vectors, {v_1, v_2,… v_m}, where each vector is denoted by v(*x*,*y*), *x* and *y* being the *x*- and *y*-co-ordinates, respectively, of the vector. Note that description (3) fully specifies the point set in figure 1 using only 12 integer values – that is, half the number required to explicitly list the co-ordinates of the points in the in extenso description in (1). Description (3) is thus a losslessly compressed encoding of description (1). Description (3) thus qualifies as an *explanation* for the structure of the point set in figure 1, precisely because it represents some of the structural regularity in this point set. If one perceives the point set in figure 1 in the way represented by description (3), then the 12 points are no longer perceived to be

arranged in a random, meaningless manner – they are now seen as resulting from the occurrence of three identical squares. Moreover, it is precisely *because* expression (3) captures this structure that it manages to convey all the information in (1) while being only roughly half the length of (1).

On the other hand, in order to generate the actual point set in figure 1 from the expression in (3), the decoder now needs to be able to interpret not only the operators P(·) and p(*x*,*y*), but also the operators T(·), V(·) and v(*x*,*y*). The decoder required to decode description (3) is therefore itself *longer* and *more complex* to describe than the decoder required to decode expression (1). The crucial question is therefore whether we save enough on the length of the encoding to warrant the resulting increase in length of the decoder. If the set of 12 points in figure 1 were the only data that we ever had to understand and the operators T(·), V(·) and v(*x*,*y*) were only of any use on this particular dataset, then the increase in the length of the decoder required to implement these extra operators would probably exceed the decrease in the length of the encoding that these operators make possible. Consequently, in this case, the parsimony principle would not predict that description (3) represented a better way of understanding the point set in figure 1 – the new encoding would just replace the specification of 8 random points in (1) with two random vectors in (3) and three randomly chosen new operators to be encoded in the decoder. However, the concepts of a vector, a vector set, and the operation of translation can be used to formulate compressed encodings of an infinite and commonly occurring class of point sets – those containing subsets related by translation. If we encode a sufficiently large sample of such point sets using translation-invariance as a compression strategy, then the saving in the lengths of the resulting encodings will more than offset the increase in the length of the decoder required to make it capable of handling translation of point sets. This illustrates that interpreting the point set in figure 1 as being composed of three identical square configurations of four points only makes sense if one is interpreting this point set in the broad context of a large (in this case, infinite) class of point sets, of which the set of points in figure 1 is an example.

The foregoing example illustrates that what we are really interested in is not just the length of an encoding but the sum of the length of the encoding and the length of the decoder required to generate the in extenso description of the encoded object from the encoding. We therefore think about descriptions of objects as being *two-part codes* in which the first part (the decoder) represents all the structural regularity in the object that it shares with all the members of a (typically large) set of other objects and the second part represents what is unique to the object and random relative to the decoder.[4] This is why we would not, for example, be interested in a "decoder" that itself consists solely of an in extenso description of the point set in figure 1 and generates this point set every time it is run with no input. In this case, the "encoding" of the data would be of length zero but, because the decoder would be of length at least equal to that of the uncompressed in extenso description of the point set, we would have no net compression and, consequently, no explanation.

3 Music analysis and data compression

If the best explanations are the shortest descriptions that account for as much data as possible in as much detail as possible, then this suggests that the goal of music analysis should be to find the shortest – but most detailed – description of as much music as possible. To illustrate this, let us consider a close musical analogue of the point-set example in figure 1 discussed above.

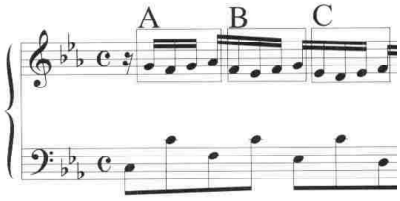

FIGURE 2. The opening notes from J. S. Bach's Prelude in C minor (BWV 871) from the second book of *Das Wohltemperierte Klavier* (1742). Patterns A, B, and C correspond, respectively, to the patterns with the same labels in figure 3 (from Meredith et al. 2002).

Figure 2 shows the beginning of J. S. Bach's Prelude in C minor (BWV 871) from the second book of *Das Wohltemperierte Klavier* (1742) and figure 3 shows a point-set representing this music, in which the horizontal dimension represents time in semiquavers and the vertical dimension represents *morphetic pitch*, an integer that encodes the pitch letter name (A–G) and octave of a note but not its alteration (…𝄫,♭,♮,♯,𝄪,…), so that, for example, D♭4, D♮4 and D♯4 all have the same morphetic pitch of 24 (Meredith 2006; 2007). The union of the three, 4-note patterns, A, B, and C, in Figure 3 could be described in an in extenso manner, on an analogy with description (1), as follows:

P(p(1,27),p(2,26),p(3,27),p(4,28),p(5,26),p(6,25),
p(7,26),p(8,27),p(9,25),p(10,24),p(11,25),p(12,26)) . (4)

This would require one to write down 24 integer co-ordinates. Alternatively, on an analogy with description (3), one could exploit the fact that the set consists of three occurrences of the same pattern at different (modal) transpositions, and describe it more parsimoniously as follows:

T(P(p(1,27),p(2,26),p(3,27),p(4,28)),V(v(4,-1),v(8,-2))) . (5)

This expression not only requires one to write down only half as many integers but also encodes some of the analytically important structural regularity in the music – namely, that the 12 points consist of three, 4-note patterns at different transpositions. Thus, by seeking a compressed encoding of the data, we have succeeded in finding a representation that gives us important information about the structural regularities in that data.

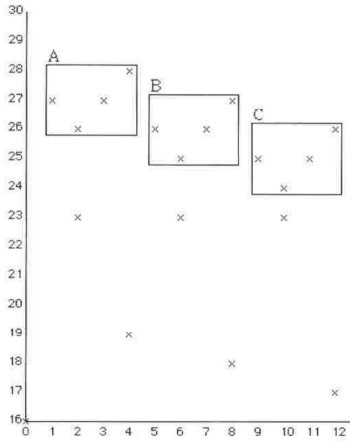

FIGURE 3. A point-set representation of the music in figure 2. The horizontal dimension represents time in semiquavers; the vertical dimension represents morphetic pitch (Meredith 2006, 2007). Patterns A, B, and C correspond, respectively, to the patterns with the same labels in figure 2. See text for further explanation (from Meredith et al. 2002).

In the particular case of figure 3, we can get an even more compact description by recognizing that the vector mapping A onto B is the same as that mapping B onto C. This means that one could represent the vector *set* V(v(4,-1),v(8,-2)) in description 5 as a vector *sequence* consisting of two consecutive occurrences of the vector v(4,-1), where the result of translating pattern A by the first vector in the sequence is itself translated by the second vector in the sequence. For example, this could be encoded as **V**(2v(4,-1)), where the emboldened **V** operator indicates that what follows is a *sequence* or *ordered* set, not an unordered set; and where we denote k consecutive occurrences of a vector, v(x,y), by kv(x,y). This would, of course, require a modification of the decoder so that it could process both vector *sequences* and the shorthand notation for sequences consisting of multiple occurrences of the same vector. As discussed above, whether or not adding this functionality to the decoder would be worthwhile depends on whether the new functionality allows for a sufficient reduction in encoding length over the whole class of musical objects that we are interested in explaining. In this particular case, since the device of

musical sequence, exemplified by the excerpt in figure 2, is commonly used throughout Western music, it would almost certainly be a good strategy to allow for the encoding of this type of structure in a compact manner. It is therefore not surprising that most psychological coding languages that have been designed for representing musical structure allow for multiple consecutive occurrences of the same interval or vector to be encoded in such a compact form (Deutsch and Feroe 1981; Meredith 2012b; Restle 1970; Simon and Sumner 1968; 1993).

4 Music-theoretical concepts that promote compact encodings of musical objects

There are a number of basic music-theoretical concepts and practices that help Western musicians and composers to encode tonal music parsimoniously and reduce the cognitive load required to process musical information.

One example of such a concept is that of a *voice*. The strategy of conceiving of music as being organized into voices substantially reduces the amount of information about note durations that has to be communicated and remembered by musicians. For the vast majority of notes in a piece of polyphonic Western music, the duration is equal to the within-voice, inter-onset interval – that is, most notes are held until the onset of the next note in the same voice. This means that, for most notes, provided we know the voice to which it belongs, we do not have to explicitly encode its duration – we only need to do so if there is a rest between it and the next note in the same voice. Grouping notes together into sequences that represent voices therefore considerably reduces the amount of information about note durations that needs to be explicitly encoded, remembered, and communicated.

The way in which pitch information is encoded in standard Western staff notation also helps to make scores more parsimonious. Key signatures, for example, remove the need to explicitly state the accidental for every note in a piece. Instead, accidentals only have to be placed

before notes whose pitches are outside the diatonic set indicated by the key signature. Since most of the notes within a single piece of Western tonal music occur within a small number of closely related diatonic sets (i.e., within a relatively limited range on the line of fifths), accidentals are typically only necessary for a small proportion of the notes in a score. Key signatures, therefore, provide a mechanism for parsimoniously encoding information about pitch names in Western tonal music.

Also, typically, Western music based on the major–minor system (or the diatonic modes) is organized into consecutive temporal segments in which each note is understood to have one of seven different basic tonal functions within the key in operation at the point where the note occurs. For example, in the major–minor system, these basic tonal functions would be {tonic, supertonic, mediant … leading note} and each could be modified or qualified by being considered flattened or sharpened relative to a diatonic major or minor scale. Staff notation capitalizes on this by providing only 7 different vertical positions at which notes can be positioned within each octave, rather than the 12 different positions that would be necessary if the pitch of each note were represented chromatically rather than in terms of its role within a seven-note scale. Again, this strategy allows for pitch information to be encoded more parsimoniously, leading to a reduction in the cognitive load on a musician reading the score.

This pitch-naming strategy leads to more parsimonious encodings by assigning simpler (shorter) encodings to pitches that are more likely to occur in the music. *Time signatures* similarly define a hierarchy of "probability" over the whole range of possible temporal positions at which a note may start within a measure. Specifically, notes are more likely to start on stronger beats.[5] In Western classical and popular music, this results in only very few possible positions within a bar being probable positions for the start (or end) of a note and the notation is designed to make it easier to notate and read notes that start at more probable positions (i.e., on stronger beats). In data compression, variable-length codes, such as the Huffman code (Huffman 1952; Cormen et al.

2009, 431–435) or Shannon–Fano code (Shannon 1948a; 1948b; Fano 1949), work in a closely analogous way by assigning shorter codes (i.e., simpler encodings) to more probable symbols or symbol strings. Huffman coding, in particular, assigns more frequent symbols to nodes closer to the root in a binary tree, which is closely analogous to tree-based representations of musical meter that assign stronger beats to higher levels in a tree structure (Lerdahl and Jackendoff 1983; Temperley 2001; 2004; 2007; Martin 1972; Meredith 1996, 214–219).

It thus seems that several features of Western staff notation and certain music-theoretical concepts have evolved in order to allow for Western tonal music to be encoded more parsimoniously.

5 Kolmogorov complexity

The work presented in this chapter is based on the central thesis that explanation *is* compression. The more compressible an object is, the less *random* it is, the *simpler* it is and the more *explicable* it is. This basic thesis was formalized by information theorists during the 1960s and encapsulated in the concept of *Kolmogorov complexity*. The *Kolmogorov complexity* of an object is a measure of the amount of *intrinsic* information in the object (Chaitin 1966; Kolmogorov 1965; Solomonoff 1964a; 1964b; Li and Vitányi 2008). It differs from the Shannon information content of an object, which is the amount of information that has to be transmitted in order to uniquely specify the object within some pre-defined set of possible objects. The Kolmogorov complexity of an object is the length in bits of the shortest possible effective (i.e., computable) description of an object, where an effective description can be thought of as being a computer program that takes no input and computes the object as its only output. In other words, the Kolmogorov complexity of an object is a measure of the complexity of the simplest process that can give rise to the object. The more

structural regularity there is in an object, the shorter its shortest possible description and the lower its Kolmogorov complexity.

Unfortunately, it is not generally possible to determine the Kolmogorov complexity of an object, as it is usually impossible to prove that any given description of the object is the shortest possible. Nevertheless, the theory of Kolmogorov complexity supports the notion of using the *length* of a description as a measure of its complexity and it supports the idea that the shorter the description of a given object, the more structural regularity that description captures. The theory has also been used to show formally that data compression is almost always the best strategy for both model selection and prediction (Vitányi and Li 2000). For some further comments on the relationship between music analysis and Kolmogorov complexity, see Meredith (2012a).

6 Music analysis and perceptual coding

As stated at the outset, the work presented here is based on the assumption that the goal of music analysis is to find the best possible explanations for musical works. This could be recast in the language of psychology by saying that music analysis aims to find the most successful *perceptual organizations* that are consistent with a given *musical surface* (Lerdahl and Jackendoff 1983).

Most theories of perceptual organization have been founded on one of two principles: the *likelihood principle* (Helmholtz 1867), which proposes that the perceptual system prefers organizations that are the most *probable* in the world; and the *simplicity principle* (Koffka 1935), which states that the perceptual system prefers the *simplest* perceptual organizations.

For many years, psychologists considered the simplicity and likelihood principles to be in conflict until Chater (1996), drawing upon the theory of Kolmogorov complexity, pointed out that the two principles are mathematically equivalent. However, Vitányi and Li (2000) showed that, strictly speaking, the predictions of the likelihood principle (which corresponds to Bayesian

inference) and the simplicity principle (which corresponds to what they call the "ideal MDL principle") are only expected to converge for individually random objects in computable distributions (Vitányi and Li 2000, 446). They state that "if the contemplated objects are nonrandom or the distributions are not computable then MDL [i.e., the simplicity principle] and Bayes's rule [i.e., the likelihood principle] may part company."

Musical objects are typically highly regular and not at all random, at least in the sense that randomness is defined within algorithmic information theory (Li and Vitányi 2008, 49ff.). Vitányi and Li's conclusions therefore seem to cast doubt on whether approaches based on the likelihood principle, commonly applied in Bayesian and probabilistic approaches to musical analysis such as those proposed by Meyer (1956), Huron (2006), Pearce and Wiggins (2012), and Temperley (2007), can ever successfully be used to discover certain types of structural regularity in musical objects such as thematic transformations or parsimonious generative definitions of scales or chords.

The approach presented in this chapter is therefore more closely aligned with models of perceptual organization based on the simplicity principle – in particular, theories of perceptual organization in the tradition of Gestalt psychology (Koffka 1935) that take the form of *coding languages* designed to represent the structures of patterns in particular domains. Theories of this type predict that sensory input is more likely to be perceived to be organized in ways that correspond to shorter descriptions in a particular coding language. Coding theories of this type have been proposed for serial patterns (Simon 1972), visual patterns (Leeuwenberg 1971), and, indeed, musical patterns (Deutsch and Feroe 1981; Meredith 2012b; Povel and Essens 1985; Restle 1970; Simon and Sumner 1968; 1993).

7 A sketch of a compression-based model of musical learning

Let us define a *musical object* to be any quantity of music, ranging from a single note through to a complete work or even a collection of works. A musical object is typically interpreted by a listener or an analyst in the context of some larger object that contains it (see figure 4). In essence, the model of musical learning presented here is as follows.[6] The analyst or listener explicitly or implicitly tries to find the *shortest* program that computes the in extenso descriptions of a set of musical objects containing:

- the object to be explained (the explanandum); and
- other objects, related to the explanandum, defining a *context* within which the explanandum is to be interpreted.

This idea is illustrated in figure 5.

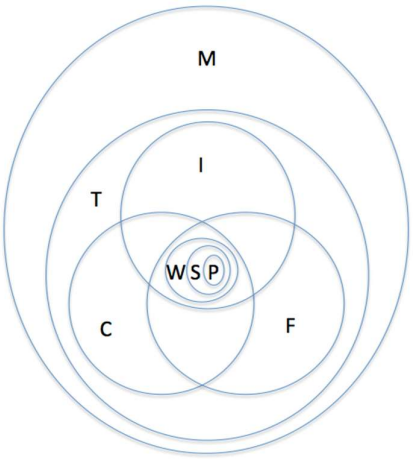

FIGURE 4. A Venn diagram illustrating various contexts in which a musical object might be interpreted. A phrase (P) could be interpreted within the context of a section (S), which could be interpreted within the context of a work (W), and so on. C = works by the same composer; F = works in the same form or genre; I = works for the same instrumentation; T = tonal music; M = all music.

The analyst and listener differ in the degree of freedom that they have to choose the context within which they interpret an object. The analyst can explicitly choose a context of closely related objects, such as other music in the same genre or by the same composer. In general, the more similar the explanandum is to the other objects in the context, the shorter its description can be, relative to that context. The listener, on the other hand, is forced to interpret the explanandum in the context of their largely implicit understanding of all the previous music they have encountered.

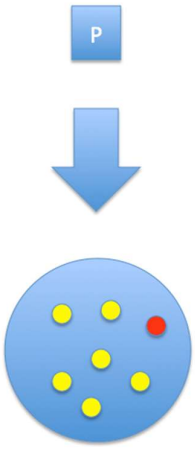

FIGURE 5. The analyst's or listener's understanding of a musical object (in red) is modeled as a program, P, that computes a set of musical objects containing the one to be explained along with other related objects (in yellow) forming a context within which the explanandum is interpreted.

Figure 6 illustrates the idea that, when the listener hears a new piece (in red), the existing explanation (i.e., program) (P) for all the music previously heard (in yellow) is minimally modified to produce a new program (P') to account for the new piece in addition to all previously encountered music. This is achieved by discovering the simplest way of interpreting as much of the material in the new piece in terms of what is already known. The perceived structure of the newly encountered musical object is then represented by the specific way in which P' computes that object. Note that P' may also generate the previously heard pieces in a way that differs from that in

which P generates these pieces, reflecting the fact that hearing a new piece may change the way that one interprets pieces that one has heard before.

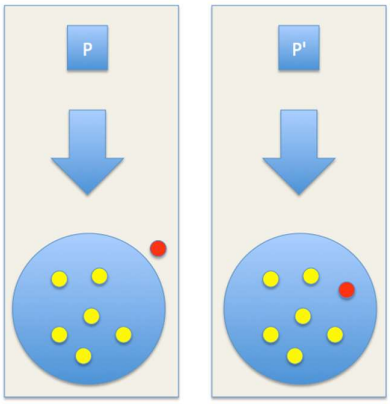

FIGURE 6. When the listener hears a new piece (in red), the existing explanation (i.e., "program") (P) for all the music previously heard is minimally modified to produce a new program (P') to account for the new piece in addition to all previously encountered music. This might be achieved by discovering the simplest way of interpreting as much of the material in the new piece in terms of what is already known.

One can speculate that P' is produced in a two-stage process. In the first stage, an attempt is made to interpret as much of the new, unfamiliar piece as possible by re-using elements and transformations that have previously been used to encode (i.e., understand) music. This will typically lead to a compact encoding of the new piece if it contains material that is related to that in previously encountered music. However, after this first stage, the global interpretation of all pieces known to the listener/analyst (including the most recently interpreted piece) may no longer be as close to optimal as it could be. In a second stage, therefore, the brain of the listener or analyst might carry out a more computationally expensive "knowledge consolidation" process in which an attempt is made to find a *globally* more efficient encoding of all music known to the individual. This might, for example, occur during sleep (see Tononi and Cirelli 2014) and might consist of a

randomized process of seeking alternative encodings of individual pieces that help to produce a more efficient global interpretation of the music known to the individual.

On this view, music analysis, perception, and learning essentially reduce to the process of compressing musical objects. This is, of course, an idealized model: for example, in practice, a listener will not have internalized a model that can account *in detail* for *all* the music they have previously heard. In other words, in reality, this learning process would probably be based on rather lossy compression.

However, it is important to stress that, even though both the analyst and the listener *aim* to find the shortest possible encodings of the music they encounter, they both usually *fail* to achieve this. As Chater (1996) points out, "the perceptual system cannot, in general, maximize simplicity (or likelihood) over all perceptual organizations…It is, nonetheless, entirely possible that the perceptual system chooses the simplest (or most probable) organization that it is able to construct" (578). This is largely a result of the limited processing and memory resources available to the perceptual system. For example, we typically describe the structure of a piece of music in terms of motives, themes, and sections, all of which are *temporally compact segments*, meaning that they are patterns that contain *all* the events that occur within a particular time span. It could well be that, for some pieces, a more parsimonious description (corresponding to a better explanation) might be possible in terms of patterns containing notes and events that are dispersed widely throughout the piece. However, listeners would normally fail to discover such patterns because their limited memories and attention spans constrain them to focus on patterns that are temporally compact (see also, Collins et al. 2011).

8 Using the model to explain individual differences

The model just sketched can be applied to understanding the emergence of differences between the ways that individuals understand the same piece. The model proposed in the previous section consists essentially of a greedy algorithm[7] that is used to construct an interpretation for a newly encountered piece that minimally modifies an existing 'program' that generates descriptions of all the pieces in a particular context set. It was proposed that this greedy approach might be supplemented by a computationally more expensive process of consolidation that attempts to find a globally more efficient encoding. Nevertheless, because such a consolidation process will not generally be capable of consistently discovering a globally optimal encoding, the way that an individual understands a given piece will generally depend not only on which pieces they already know, but also on the *order* in which these pieces were encountered. This implication could fairly straightforwardly be tested empirically.

A rather crude version of the foregoing model has been implemented in an algorithm called SIATECLearn. The SIATECLearn algorithm is based on the geometric pattern discovery algorithm, SIATEC, proposed by Meredith and colleagues (2002). The SIATEC algorithm takes as input a set of points called a *dataset* and automatically discovers all the translationally related occurrences of maximal repeated patterns in the dataset. If the dataset represents a piece of music, with each point representing a note in pitch-time space, then two patterns in this space related by translation correspond to two statements of the same musical pattern, possibly with transposition. We say a pattern P is *translatable* within a dataset D if there exists a vector, v, such that P translated by v gives a pattern that is also in D. A translatable pattern is *maximal* for a given vector, v, in a dataset D, if it contains *all* the points in the dataset that can be mapped by translation by v onto other points in the dataset. The maximal translatable pattern (MTP) for a vector v in a dataset D, which we can denote by $MTP(v, D)$, can also be thought of as being the intersection of the dataset D and the dataset D translated by $-v$. That is,

$$MTP(v, D) = D \cap (D - v). \quad (6)$$

For each (non-empty) maximal translatable pattern, *P*, in a dataset, SIATEC finds *all* the occurrences of *P*, and outputs this occurrence set of *P*. Such an occurrence set is called the *translational equivalence class* (*TEC*) of *P* in *D*, denoted by $TEC(P, D)$, because it contains all the patterns in the dataset that are translationally equivalent to *P.* That is,

$$TEC(P, D) = \{Q \mid Q \subseteq D \wedge (\exists v | Q = P + v)\}. \quad (7)$$

SIATEC therefore takes a dataset as input and outputs a collection of TECs, such that each TEC contains all the occurrences of a particular maximal translatable pattern.

An algorithm called SIATECCompress (Meredith 2013b; 2015; 2016) runs SIATEC on a dataset, then sorts the found TECs into decreasing order of "quality." Given two TECs, the one that results in the better compression (in the sense of expressions (4) and (5) above) is deemed superior. If both TECs give the same degree of compression, then the one whose pattern is spatially more compact is considered superior. SIATECCompress then scans this list of occurrence sets and computes an encoding of the input dataset in the form of a set of TECs that, taken together, account for or *cover* the entire input dataset.

SIATECLearn runs SIATECCompress, but also stores the patterns it finds on each run and will preferably re-use these patterns rather than newly found ones on subsequent runs of the algorithm. Thus, when SIATECLearn is run on the 12-point pattern on the left in figure 7, it "interprets" the dataset as being constructed from three occurrences of the square pattern shown. This square pattern is therefore stored in its "long-term" memory. When the algorithm is subsequently run on the 10-point dataset on the right, it prefers to use the stored square pattern rather than any of the patterns that it finds in this newly encountered dataset; it interprets the new dataset as containing two occurrences of the square pattern along with two extra points.

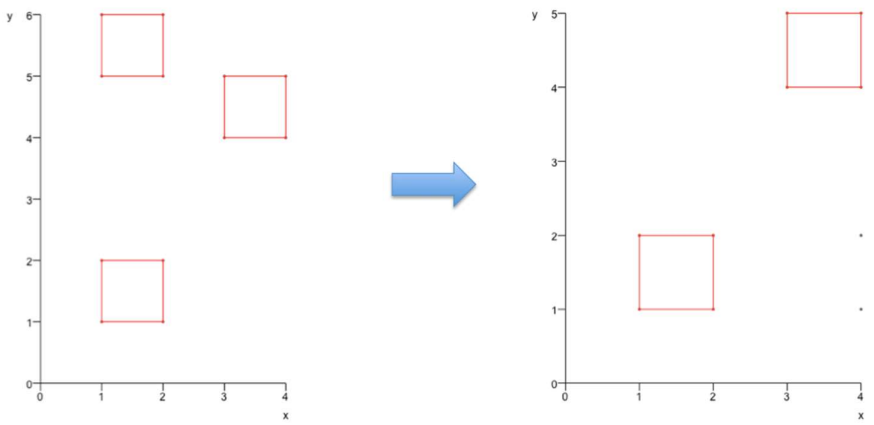

FIGURE 7. Output of SIATECLearn when presented first with the dataset on the left and then with the dataset on the right.

Conversely, when SIATECLearn is first presented with the 10-point dataset, it interprets the dataset as being composed from 5 occurrences of the two-point vertical line configuration shown on the left in figure 8. This pattern is then stored in long-term memory, so that, when the algorithm is subsequently presented with the 12-point dataset, it interprets this set as consisting of 6 occurrences of this vertical line rather than 3 occurrences of the square pattern. This very simple example illustrates how the way in which objects are interpreted can depend on the order in which they are presented.

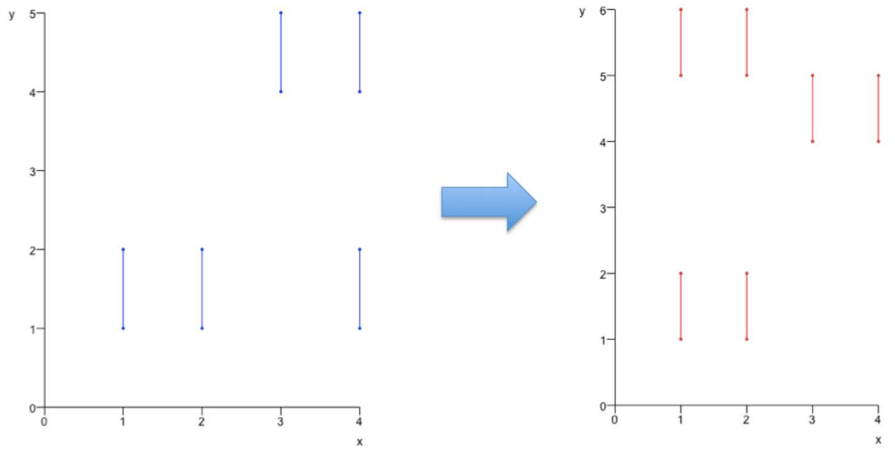

FIGURE 8. Output of SIATECLearn when presented first with the dataset on the left and then with the dataset on the right.

<1>COSIATEC: Music analysis by point-set compression

Given the concept of a TEC, as defined in (7) above, we can define the covered set, $CS(T)$, of a TEC T to be the union of all the patterns in T. That is

$$CS(T) = \bigcup_{P \in T} P. \tag{8}$$

COSIATEC (Meredith et al. 2003; Meredith 2013b; 2015; 2016) is a greedy compression algorithm based on SIATEC. The algorithm takes a dataset as input and computes a set of TECs that collectively cover this dataset in such a way that none of the TECs' covered sets intersect. It also attempts to choose this set of TECs so that it minimizes the length of the output encoding. The basic idea behind the algorithm is sketched in the pseudo-code in figure 9.

```
COSIATEC(S)
   while S is not empty
      Find the best TEC, T, using SIATEC
      Add T to the encoding, E
      Remove the points covered by T from S
   return the encoding E
```

FIGURE 9. The COSIATEC algorithm.

As shown in figure 9, the COSIATEC algorithm first finds the "best" TEC in the output of SIATEC for the input dataset, *S*. The best TEC is the one that produces the best compression. This means that it is the one that has the best *compression factor*, which is the ratio of the number of points in its covered set (as defined in (8)) to the sum of the number of points in one occurrence of the TEC's pattern and the number of occurrences minus 1. The reasoning behind this is that a TEC can be compactly encoded as an ordered pair, (*P,V*), where *P* is one occurrence in the TEC and *V* is the set of vectors that map *P* onto all the other occurrences of *P* in the dataset. The number of vectors in *V* is therefore equal to the number of occurrences of *P* minus 1. The length of an in extenso encoding of a TEC's covered set in terms of points is simply $|CS(T)|$ as defined in (8). Each vector in *V* has approximately the same information content as a point in *P,* so the length of an ordered pair encoding of a TEC, (*P*,*V*), in terms of points is approximately |*P*|+|*V*|. The compression factor is the ratio of the length of the in extenso encoding to the length of the compressed encoding. Thus, the compression factor of a TEC, *T* = (*P*,*V*), denoted *CF*(*T*), can be defined as

$$CF(T) = \frac{|CS(T)|}{|P| + |V|}.$$

If two TECs have the same compression factor, then COSIATEC chooses the TEC in which the first occurrence of the pattern is the more *compact*: the *compactness* of a pattern is the ratio of the number of points in the pattern to the number of dataset points in the bounding box of the pattern. The rationale behind this heuristic is that patterns are more likely to be noticeable if the region of pitch-time space that they span does not also contain many 'distractor' points that are not in the pattern. These heuristics for evaluating the quality of a TEC are discussed in more detail by Meredith and colleagues (2002), Meredith (2015), and Collins and co-authors (2011).

As shown in figure 9, once the best TEC, *T*, has been found for the input dataset, *S*, this TEC is added to the encoding (*E*) and the covered set of *T*, *CS*(*T*), is removed from *S*. Once the

covered set of *T* has been removed from *S*, the process is repeated, with SIATEC being run on the new *S*. The procedure is repeated until *S* is empty, at which point *E* contains a set of TECs that collectively cover the entire input dataset. Moreover, because the TEC that gives the best compression factor is selected on each iteration, *E* is typically a *compact* or *compressed* encoding of *S*. COSIATEC typically produces encodings that are more compact than those produced by SIATECCompress.

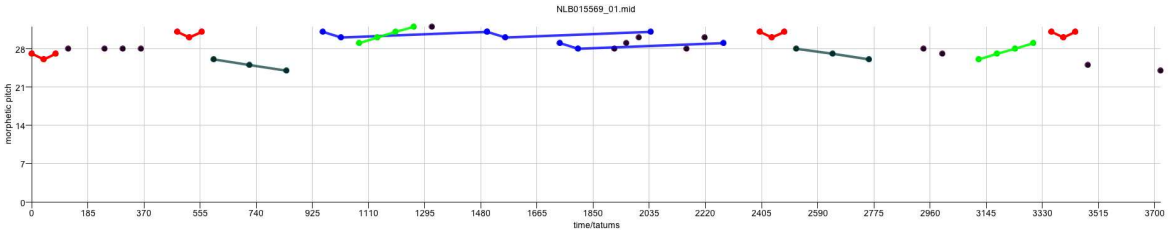

FIGURE 10. The set of TECs computed by COSIATEC for a short Dutch folk song, "*Daar zou er en maagdje vroeg opstaan*" (file number NLB015569 from the Nederlandse Liederen Bank, http://www.liederenbank.nl). Courtesy of Peter van Kranenburg.

Figure 10 shows the output of COSIATEC for a short Dutch folk song. The complete piece can be encoded as the union of the covered sets of 5 TECs. In figure 10, each TEC is drawn in a different color. The first TEC, drawn in red, consists of the occurrences of a three-note, lower-neighbor-note figure. This TEC has the best compression factor of any TEC for a maximal translatable pattern in this dataset. After these three-note patterns have been removed from the piece, the next best TEC is the one drawn in light green in figure 10, namely the two occurrences of the four-note, rising scale segment. The fifth TEC consists of the 14 occurrences of a single unconnected point in figure 10. These are the points (notes) that are left over after removing the sets of repeated patterns that give the best compression factor. This final set of "residual" points,

which cannot be compressed by the algorithm, is essentially seen by the algorithm as being random "noise" that it cannot "explain."

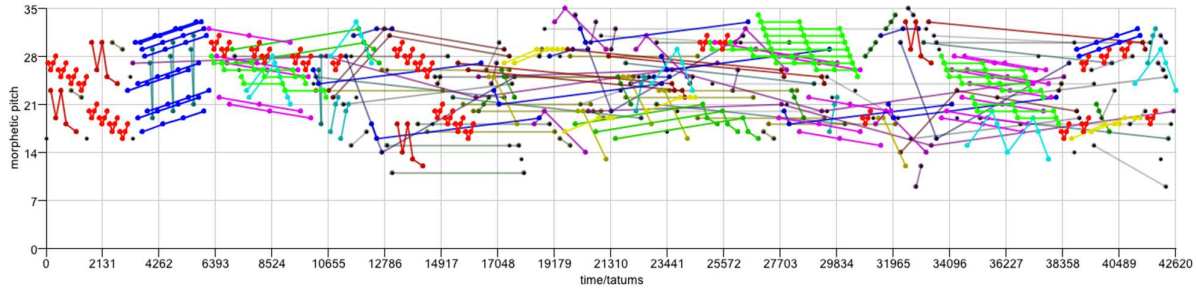

FIGURE 11. Analysis generated by COSIATEC of J. S. Bach's Prelude in C minor (BWV 871) from the second book of *Das Wohltemperierte Klavier* (1742). All the occurrences of a given pattern are drawn in a single color. The first TEC generated, in red, is the one that has the highest compression factor over the whole dataset. The overall compression factor of this analysis is 2.3 and the residual point set, containing notes that the algorithm does not re-express in a compact form, contains 3.61% of the notes in the piece (corresponding to 25 out of 692 notes).

Figure 11 shows the analysis generated by COSIATEC for a more complex piece of music, the Prelude in C minor (BWV 871) from book 2 of J. S. Bach's *Das Wohltemperierte Klavier*. Note that the first TEC (in red) generated by COSIATEC (i.e., the one that results in the most compression over the whole dataset) is precisely the four-note pattern shown in figure 2 above.

9 Evaluating music analysis algorithms

In the introduction to this chapter, it was proposed that, when given two or more different analyses of the same piece of music (or, more generally, musical object), it may be possible to determine which of the analyses is the best for carrying out certain objectively evaluable tasks. It is similarly

possible to evaluate algorithms that compute analyses by comparing how well the generated analyses allow certain tasks to be performed.

In a recent paper (Meredith 2015), the point-set compression algorithms, COSIATEC and SIATECCompress, were compared on a number of different tasks with a third greedy compression algorithm proposed by Forth and Wiggins (2009) and Forth (2012). The algorithms were evaluated on three tasks: folk song classification, discovery of repeated themes and sections, and discovery of fugal subject and counter-subject entries. Although no obvious correlation was found between compression factor and performance on these tasks, COSIATEC achieved both the best compression factor (around 1.6) and the best classification success rate (84%) on the folk-song classification task. The pattern-discovery task on which the algorithms compared in this study were evaluated consisted of finding the repeated themes and sections identified in the JKU Patterns Development Database, a collection of five pieces of classical and baroque music, each accompanied by "ground-truth" analyses by expert musicologists (Collins 2013). The output of each algorithm was compared with these analyses. I have argued (Meredith 2015, 263–265) that these "ground-truth" analyses are not satisfactory for at least two reasons: first, the musicologists on whose work the ground-truth analyses are based did not consistently identify all occurrences of the patterns that they considered to be worth mentioning; and second, there are patterns that are noticeable and important that the musicologists who created the ground-truth analyses failed to mention. Indeed, the tested algorithms discovered not only structurally salient patterns that the analysts omitted to mention but also exact occurrences of the ground-truth patterns that are not recorded in the ground-truth analyses. Figure 12 shows some examples of structurally important patterns in a fugue by J. S. Bach that were not recorded in the "ground-truth" analyses used for evaluation.

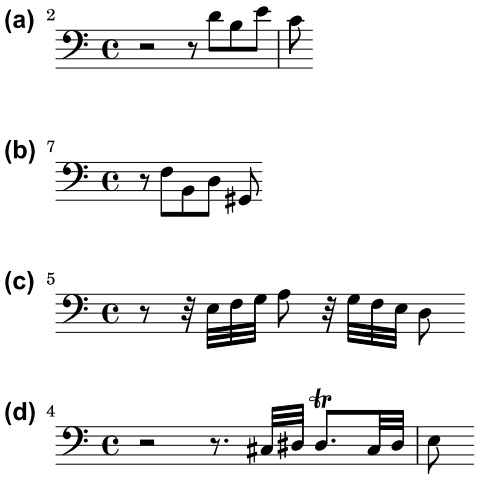

FIGURE 12. Examples of noticeable and/or important patterns in Bach's Fugue in A minor (BWV 889), that were discovered by the algorithms tested by Meredith (2015) but were not recorded in the "ground-truth" analyses in the JKU Patterns Development Database used for evaluation. Patterns (a), (b), and (d) were discovered by COSIATEC. Patterns (c) and (d) were discovered by SIATECCompress.

Notwithstanding the foregoing methodological issues with this task, it was found that SIATECCompress performed best on average, achieving an average F_1 score of about 50% over the five pieces in the corpus. However, COSIATEC, achieved F_1 scores of 71% and 60% on the pieces by Beethoven and Mozart, respectively; and Forth's algorithm performed substantially better than the other algorithms on a fugue by Bach. There was therefore no algorithm that consistently performed best on this task.

On the fugal analysis task, the algorithms performed rather less well than on the other evaluation tasks. COSIATEC and SIATECCompress achieved a mean recall of around 60% over the 24 fugues in the first book of J. S. Bach's *Das Wohltemperierte Klavier*. However, COSIATEC's precision on this task was much lower (around 10%). Overall, the best performing algorithm was SIATECCompress that achieved an F_1 score of around 30% on this fugal analysis task.

In the study just discussed, the performance of the SIA-based compression algorithms on the folk-song classification task was compared with that of the general-purpose text compression algorithm, bzip2 (Seward 2010). On this task, bzip2 achieved a much higher average compression factor (3.5) but a much lower classification success rate (12.5%) than the SIA-based algorithms. At first sight, this might be interpreted as evidence against the basic hypothesis that shorter descriptions correspond to better explanations. In a later study, Corentin Louboutin and I therefore explored in more depth whether general-purpose compression algorithms could be used for music analysis, by comparing three general-purpose compression algorithms with COSIATEC on two music-analytical tasks (Louboutin and Meredith 2016). The general-purpose algorithms compared included the Burrows–Wheeler algorithm (Burrows and Wheeler 1994), Lempel–Ziv-77 (LZ77) (Ziv and Lempel 1977) and Lempel–Ziv-78 (LZ78) (Ziv and Lempel 1978). This study confirmed that, in order to achieve good results, the type of representation used for the music has to be appropriate for the compression algorithm used. Thus, COSIATEC, which discovers maximal repeated patterns in point sets, was unaffected by the order in which the notes were sorted in the input files. However, LZ77 discovers repeated substrings in a sequence of symbols and these substrings, consisting of sequences of contiguous symbols in the original string representation, only correspond to sequences of contiguous notes in a voice when the notes in the music are presented to the algorithm a voice at a time. If the notes are presented a chord at a time (i.e., sorting the notes first by pitch and then by onset), then we should not expect LZ77 to be capable of finding repeated melodic themes. Our results confirmed this; when the algorithms were used on the fugal analysis task described above, the F_1 score for the LZ77 algorithm doubled when the notes were first sorted so that the algorithm was presented with the music a voice at a time rather than a chord at a time. On this task, we also found a strong correlation between compression factor and F_1 score, supporting the general notion that shorter descriptions represent better explanations.

On the folk-song classification task, we were able to improve on the performance of COSIATEC by using 8 different representations in combination, with LZ77 being used to calculate normalized compression distances for seven of these and COSIATEC being used for the last one. In this way, we succeeded in achieving a classification success rate of over 94% using an 8-nearest-neighbor classification algorithm, compared with 85% for COSIATEC alone.

10 Applying a compression-driven approach to the analysis of musical audio

The main concern in this chapter has been with explaining "musical objects" by discovering losslessly compressed descriptions of these objects. The basic scheme is that one takes an in extenso encoding of such an object and then attempts to find a short algorithm that generates that in extenso encoding as its only output. The encoding could be on any level of granularity and could represent any quantity of music in any possible domain in which a musical object might be manifested – for example, an image of a score, a symbolic encoding of a score, an audio recording or a video recording. In the examples and evaluations presented above, the focus has been on musical objects that are symbolic encodings of scores. In such cases, one can realistically hope to be able to produce losslessly compressed descriptions in which we are required to consider only a very small proportion of the information in the object to be "random" or "noise." On the other hand, if one were concerned with explaining the structure of a digital audio recording of a performance of a piece produced by human performers playing from a score, then one would expect the compression factors achievable to be lower and one would expect to have to be satisfied with considering a larger proportion of the information in the object as being "noise." This is because the detailed structure of such a recording depends not only on the score from which the players are performing, but also many other factors that are perhaps harder to model, such as the acoustics of the space in which the recording was made, the precise nature of the instruments used

and, most importantly, the players themselves and their own particular ways of interpreting the score.

11 Summary

In this chapter, I have proposed that the goal of music analysis should be to find the "best" ways of understanding musical objects and that two different analyses of the same musical object can be compared objectively by determining whether one of them allows us to more effectively perform some specific set of tasks. I have also explored the hypothesis that, for all tasks that require an understanding of how a musical object is constructed, the best ways of understanding that object are those that are represented by the shortest possible descriptions of the object. I have briefly outlined how this hypothesis relates to the theory of Kolmogorov complexity and to coding theory models of perception. I have also briefly sketched how these ideas can form the basis of a theory of musical learning that can potentially explain aspects of music cognition such as individual differences. Finally, I briefly described the COSIATEC point-set compression algorithm and reviewed the results of some experiments in which it and other related algorithms have been used to automatically carry out musical tasks such as folk-song classification and thematic analysis. The results achieved in these experiments generally support the idea that the knowledge necessary to be able to successfully carry out advanced musicological tasks can largely be acquired simply by compressing in extenso representations of musical objects. Moreover, some of the results clearly indicate a correlation between compression factor and success on musicological tasks. However, these experiments also show that performance on such tasks depends heavily both on the specific types of redundancy exploited by the compression algorithm used to generate the compressed encodings and on the precise form of the in extenso representations used as input to these compression-based learning methods.

12 Acknowledgements

The work reported in this chapter was carried out as part of the EU collaborative project, "Learning to Create" (Lrn2Cre8). The project Lrn2Cre8 acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

13 References

Bouchier, E. S. 1901. *Aristotle's Posterior Analytics*. Oxford: Oxford University Press.

Burrows, M., and D. J. Wheeler. 1994. A Block-Sorting Lossless Data Compression Algorithm. Palo Alto, CA.: Digital Systems Research Center (now HP Labs). Technical Report SRC 124.

Chaitin, G. J. 1966. On the Length of Programs for Computing Finite Binary Sequences. *Journal of the Association for Computing Machinery* 13 (4): 547–569.

Chater, N. 1996. Reconciling Simplicity and Likelihood Principles in Perceptual Organization. *Psychological Review* 103 (3): 566–581.

Collins, T. 2013. JKU Patterns Development Database. https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip. Accessed January 21, 2016.

Collins, T., R. Laney, A. Willis, and P. H. Garthwaite. 2011. Modeling Pattern Importance in Chopin's *Mazurkas. Music Perception* 28 (4): 387–414.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2009. *Introduction to Algorithms*. 3rd edition. Cambridge, MA.: MIT Press.

Deutsch, D., and J. Feroe. 1981. The Internal Representation of Pitch Sequences in Tonal Music. *Psychological Review* 88 (6): 503–522.

Fano, R. M. 1949. The Transmission of Information. Technical report no. 65 March 17. Cambridge, MA.: Research Laboratory of Electronics, MIT.

Forth, J. C. 2012. Cognitively-Motivated Geometric Methods of Pattern Discovery and Models of Similarity in Music. PhD thesis. Department of Computing, Goldsmiths, University of London, UK.

Forth, J., and G. A. Wiggins. 2009. An Approach for Identifying Salient Repetition in Multidimensional Representations of Polyphonic Music. In *London Algorithmics 2008: Theory and Practice*, edited by J. Chan, J. W. Daykin, and M. S. Rahman, 44–58. London: College Publications.

Halpern, A. R. 2003. Cerebral Substrates of Musical Imagery. In *The Cognitive Neuroscience of Music,* edited by I. Peretz and R. J. Zatorre, Chapter 15. Oxford: Oxford University Press. DOI: 10.1093/acprof:oso/9780198525202.001.0001.

Helmholtz, H. L. F. 1867. *Handbuch der physiologischen Optik.* Leipzig: Leopold Voss.

Huffman, D. A. 1952. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE*, September: 1098–1101.

Huron, D. 2006. *Sweet Anticipation: Music and the Psychology of Expectation*. Cambridge, MA.: MIT Press.

Koffka, K. 1935. *Principles of Gestalt Psychology*. New York: Harcourt Brace.

Kolmogorov, A. N. 1965. Three Approaches to the Quantitative Definition of Information. *Problems of Information Transmission* 1 (1): 1–7.

Leeuwenberg, E. L. J. 1971. A Perceptual Coding Language for Visual and Auditory Patterns. *American Journal of Psychology* 84 (3): 307–349.

Lerdahl, R., and R. Jackendoff. 1983. *A Generative Theory of Tonal Music*. Cambridge, MA.: MIT Press.

Li, M., and P. M. B. Vitányi. 2008. *An Introduction to Kolmogorov Complexity and Its Applications*. 3rd edition. New York: Springer.

Louboutin, C., and Meredith, D. 2016. Using General-Purpose Compression Algorithms for Music Analysis. *Journal of New Music Research* 45 (1): 1–16.

Martin, J. G. 1972. Rhythmic (hierarchical) versus Serial Structure in Speech and Other Behavior. *Psychological Review* 79 (6): 487–509.

Meredith, D. 1996. The Logical Structure of an Algorithmic Theory of Tonal Music. Unpublished thesis. http://www.titanmusic.com/papers/public/thesis1996.pdf. Accessed May 15, 2017.

Meredith, D. 2006. The *ps13* Pitch Spelling Algorithm. *Journal of New Music Research* 35 (2): 121–159.

Meredith, D. 2007. Computing Pitch Names in Tonal Music: A Comparative Analysis of Pitch Spelling Algorithms. PhD thesis. University of Oxford.

Meredith, D. 2012a. Music Analysis and Kolmogorov Complexity. *Proceedings of the 19th Colloquio d'Informatica Musicale (XIX CIM)*, Trieste, Italy.

Meredith, D. 2012b. A Geometric Language for Representing Structure in Polyphonic Music. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, Porto, Portugal: 133–138.

Meredith, D. 2012c. A Compression-Based Model of Musical Learning. *DMRN+7: Digital Music Research Network One-day Workshop 2012,* December 18, Queen Mary University of London.

Meredith, D. 2013a. Analysis by Compression: Automatic Generation of Compact Geometric Encodings of Musical Objects. *The Music Encoding Conference 2013*. May 22–24, Mainz Academy for Literature and Sciences, Mainz, Germany.

Meredith, D. 2013b. COSIATEC and SIATECCompress: Pattern Discovery by Geometric Compression. *Music Information Retrieval Evaluation Exchange (Competition on "Discovery of Repeated Themes & Sections") (MIREX)*, Curitiba, Brazil.

Meredith, D. 2015. Music Analysis and Point-Set Compression. *Journal of New Music Research* 44 (3): 245–270.

Meredith, D. 2016. Analysing Music with Point-Set Compression Algorithms. In *Computational Music Analysis,* edited by D. Meredith, 335–366. Cham, Switzerland: Springer.

Meredith, D., K. Lemström, and G. A. Wiggins. 2002. Algorithms for Discovering Repeated Patterns in Multidimensional Representations of Polyphonic Music. *Journal of New Music Research* 31 (4): 321–345.

Meredith, D., K. Lemström, and G. A. Wiggins. 2003. Algorithms for Discovering Repeated Patterns in Multidimensional Representations of Polyphonic Music. *Proceedings of the Cambridge Music Colloquium*, University of Cambridge.

Meyer, L. B. 1956. *Emotion and Meaning in Music*. Chicago: Chicago University Press.

Pearce, M., and G. A. Wiggins. 2012. Auditory Expectation: The Information Dynamics of Music Perception and Cognition. *Topics in Cognitive Science* 4: 625–652.

Povel, D.-J., and P. Essens. 1985. Perception of Temporal Patterns. *Music Perception* 2 (4): 411–440.

Restle, F. 1970. Theory of Serial Pattern Learning: Structural trees. *Psychological Review* 77 (6): 481–495.

Rissanen, J. 1978. Modeling by Shortest Data Description. *Automatica* 14: 465–471.

Seward, J. 2010. bzip2 version 1.0.6, released 20 September 2010. http://www.bzip.org. Accessed April 19, 2014.

Shannon, C. E. 1948a. A Mathematical Theory of Communication. *The Bell System Technical Journal* 27 (3): 379–423.

Shannon, C. E. 1948b. A Mathematical Theory of Communication. *The Bell System Technical Journal* 27 (4): 623–656.

Simon, H. A. 1972. Complexity and the Representation of Patterned Sequences of Symbols. *Psychological Review* 79 (5): 369–382.

Simon, H. A., and R. K. Sumner. 1968. Pattern in Music. In *Formal Representation of Human Judgment,* edited by B. Kleinmuntz. New York: Wiley.

Simon, H. A., and R. K. Sumner 1993. Pattern in music. In *Machine Models of Music,* edited by S. M. Schwanauer and D. A. Levitt, 83–110. Cambridge, MA.: MIT Press.

Solomonoff, R. J. 1964a. A Formal Theory of Inductive Inference, part I. *Information and Control* 7 (1): 1–22.

Solomonoff, R. J. 1964b. A Formal Theory of Inductive Inference, part II. *Information and Control* 7 (2): 224–254.

Temperley, D. 2001. *The Cognition of Basic Musical Structures*. Cambridge, MA.: MIT Press.

Temperley, D. 2004. An Evaluation System for Metrical Models. *Computer Music Journal* 28 (3): 28–44.

Temperley, D. 2007. *Music and Probability*. Cambridge, MA.: MIT Press.

Tononi, G., and C. Cirelli. 2014. Sleep and the Price of Plasticity: From Synaptic and Cellular Homeostasis to Memory Consolidation and Integration. *Neuron* 81 (1): 12–34.

Vereshchagin, N. K., and P. M. B. Vitányi. 2004. Kolmogorov's Structure Functions and Model Selection. *IEEE Transactions on Information Theory* 50 (12): 3265–3290.

Vitányi, P. M. B., and M. Li. 2000. Minimum Description Length Induction, Bayesianism, and Kolmogorov complexity. *IEEE Transactions on Information Theory* 46 (2): 446–464.

Ziv, J., and A. Lempel. 1977. A Universal Algorithm for Sequential Data Compression. *IEEE Transactions on Information Theory* 23 (3): 337–343.

Ziv, J., and A. Lempel. 1978. Compression of Individual Sequences via Variable-Rate Coding. *IEEE Transactions on Information Theory* 24 (5): 530–536.

---

[1] See, for example, *Chap.* XXV of *Book* 1 of Aristotle's *Posterior Analytics* (Bouchier 1901, 66).

[2] Kolmogorov introduced the field of non-probabilistic statistics at a conference in Tallinn, Estonia, in 1973 and in a talk at the Moscow Mathematical Society in 1974 (Li and Vitányi 2008, 405). Unfortunately, these talks were never published in written form.

[3] See Simon and Sumner (1968, 1993) for a similar use of the term 'in extenso' in the context of music representations.

[4] For a more technical discussion of two-part codes, see Vitányi and Li (2000, 447).

[5] See Temperley (2007, chapter 3) for a model of rhythm and meter perception based on the idea that simpler meters are more probable and events are more likely to occur on stronger beats.

[6] This model was originally described by Meredith (2012c, 2013a).

[7] A *greedy algorithm* attempts to solve an optimization problem by always choosing the locally best option at each decision point in the construction of a solution. This does not always produce a globally optimal solution, but for some problems it does (e.g., activity selection, the construction of a Huffman code). For more details, see Cormen et al. (2009, 414–450).