HIERARCHICAL MARKOV MODELING FOR GENERATIVE MUSIC

Chris Thornton

COGS/Informatics
University of Sussex, Brighton, BN1 9QH, UK
c.thornton@sussex.ac.uk

ABSTRACT

The paper describes a hierarchical Markov modeling strategy that offers the advantages of a statistical approach without constraining the level of analysis. The method can mediate music generation from sample compositions. Some illustrative examples are described.

1. INTRODUCTION

A well-established genre in computer music involves generation of novel compositions on the basis of a model derived from existing compositions. This can be done on a statistical basis, e.g., by examining the relative probabilities of musical subsequences and generating a piece which conforms to the observed distribution. This often involves derivation of some form of Markov model, e.g., [1], and use of transition probabilities for purposes of building up a novel sequence.

A problem with such approaches is the difficulty of choosing the level of analysis [2]. A statistical model is necessarily constructed in terms of whatever entity is selected for sampling; the level of analysis is then fixed in advanced. If we construct a statistical model on the basis of note-transition probabilities, the model obtained will tend to represent note-transition *structures*. Structures at other scales may not be represented.[1] Music generated from the model may conform to observed structure at one level while deviating from it at another. The well known 'wandering melody' effect, exhibited by some Markov-chained music, is a case in point.

Ideally, statistical modeling should have some way of discovering and adapting to different levels of analysis, so as to incorporate the kinds of hierarchical functionality associated with, e.g., the viewpoints framework of [3] and analytic applications of classification [4; 5], style analysis [6] and segmentation [7]. But if this involves introduction of domain knowledge, the question is raised of how we can be confident in our choice of musical descriptors.

[1] In principle, hidden Markov models have the potential to capture dependencies at any scale. However, in practice, capture of long-range dependencies underpinning hierarchical-structure requires processing of prohibitive amounts of data [2].

The approach described by the present paper overcomes this difficulty by removing the element of choice. Hierarchical knowledge is not imposed on the statistical process. Rather, it emerges, much in the way that grammatical structure emerges in hierarchical dictionary methods [8]. The method decomposes modeling into two steps. In the initial step, an informationally optimal hierarchical model (based on rewrite rules) is derived. In the second step, a Markov model is derived for each level of the hierarchy. Structure is then modeled at multiple levels of analysis, but with statistical effects at any particular level captured in the ordinary way. Generative use of the model then proceeds on the basis of top-down symbol expansion, with disjunctive choices being resolved using the relevant Markov model.

2. THEORETICAL PRELIMINARIES

A hierarchical model for a sequence *S* is taken to be (the equivalent of) a system of rewrite rules capable of generating *S* from a single symbol. Thus, the rules

$$X \rightarrow Y, Z$$
$$Y \rightarrow A, B/C$$
$$Z \rightarrow D$$

form a hierarchical model for the sequences *A B C* and *A C D* because both can be generated from *X*.[2]

Although the task of finding the simplest model is noncomputable in general [9] in the particular case of a hierarchical rewrite model, there is the possibility of breaking the task down into tractable steps. The task of finding the smallest hierarchical model can be decomposed into the *recursive* operation of finding the rule encoding which maximizes information-content per symbol.

Information content per symbol of a hierarchical sequence model may be defined as follows.

$$\frac{\sum_{i=1}^{l(S)} log(k) - log(n_i)}{|E| + l(E)}$$

[2] Note, use of the disjunctive element *B/C* in the second rule, representing *B or C*.

Here, $l(S)$ is the length of the original sequence, k is the size of alphabet, n_i is the number of alternatives specified by the encoding for the i'th element of the sequence, $|E|$ is the number of symbols used in the encoding and $l(E)$ is its length. The formula sums the information contents of individual elements of the sequence, discounting any uncertainties introduced through disjunctive rules and dividing by the total number of symbols involved.

As an illustration, let 'Y Z' be considered to be an encoding of 'A C D' using the rules for Y and Z shown above. Assuming an alphabet of 8 symbols and noting that the second constituent in the rule for Y has two alternatives, we then evaluate the information-per-symbol of the model to be

$$\frac{(\log(8) - \log(1)) + (\log(8) - \log(2)) + (\log(8) - \log(1))}{4+2}$$

or approximately 1.333 bits.

Using this measure, derivation of an informationally optimal hierarchical model can be undertaken incrementally. At step one, an encoding is obtained that maximizes information-content per symbol.[3] At step two, this encoding is substituted for the original sequence and the procedure is re-applied. (This produces an encoding of the encoding.) At step three, the procedure is repeated, producing an encoding of the encoding of the encoding. The process continues recursively until $l(E) = 1$.

3. WORKED EXAMPLE

Table 1 illustrates the process in action. The sequence used in the example is 'a b c a d c a e': this is shown in the bottom cell of the first column. Assuming an alphabet of 8 characters, information per symbol is 3 bits. At step one, the procedure finds the set of rules that yield an informationally optimal encoding. These are shown in the middle column of the third row. The associated encoding is to the left. It consists of just three symbols but the rules involved use a further five. Both symbols of the encoding introduce uncertainty through disjunction. The uncertainty-increase and information-compression effects balance out, leaving information-per-symbol at the original level of 3 bits.

At the next stage, derivation of a single rule (second row, middle column) leads to the encoding of row two, which increases information-per-symbol to 4.8 bits. Derivation of a single rule (middle cell, top row) then produces the encoding shown in the left cell of the top row. The process then terminates. Taking into account the 24 bits of information from the original sequence and the three symbols involved in the encoding (R, Z and Y), information per symbol at termination is 8 bits.

[3] Feasible methods are described by [8].

Sequence	Rules derived	Inf. per symbol
R	R → Z Y	8 bits
Z Y	Z → X/Y Y	4.8 bits
X Y Y	X → a b/d/e Y → c a d/e	3 bits
a b c a d c a e		3 bits

Table 1. Informational hierarchy derivation

The hierarchical model derived can be used directly for generative purposes if we wish. Taking the top-level symbol from the hierarchy, we recursively expand it using the rules of the system (making random choices as necessary) until no further progress is possible. This is illustrated in Table 2. At step one, a sequence is constructed consisting of just the model's top symbol: R. Invoking the rule from row one, the sequence can then be expanded as shown in row two. Invoking rules from row two and making random choices where necessary, it then takes the form shown in row three. Finally, on the basis of rules in row three, we obtain the sequence of row four.

Sequence	Rules applied
R	R → Z Y
Z Y	Z → X/Y Y Y → c a d/e
X Y c a d	X → a b/d/e Y → c a d/e
a b c a d c a d	

Table 2. Hierarchical symbol expansion

4. THE MARKOV STEP

In the proposed approach, a Markov model is derived for each derived sequence, and thus each level of the hierarchy. These are brought into play at the generative stage. Top-down symbol expansion proceeds in the usual way except that wherever there is a choice between alternative expansions, this is resolved not at random but in conformity with the relevant Markov model. The screenshot of Figure 1 (generated using an applet-based implementation of the method)[4] illustrates this procedure applied to the sequence 'c c d b c d e e f e d c d c b c'. This represents the first 16 notes from the melody of the British national anthem 'God Save the Queen'. (Non-primitive symbols in this figure are randomly generated strings.)

The general structure of the representation is similar to

[4] This can be accessed from www.christhornton.eu/demos/reconstituted-music-examples.html. The example uses the 'queen1' input set.

```
~~~~~ nada ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

nada --> ws ws

P(ws|ws) = 1.0

~~~~~ ws ws ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

ws --> wc Tc/naba

P(naba|wc) = 0.5            P(wc|naba) = 1.0
P(Tc|wc) = 0.5

~~~~~ wc naba wc Tc ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

wc --> Xx/c YZ
naba --> YZ na

P(YZ|c) = 1.0               P(Tc|YZ) = 0.333
P(YZ|YZ) = 0.33             P(Xx|na) = 1.0
P(na|YZ) = 0.33             P(YZ|Xx) = 1.0

~~~~~ c YZ YZ na Xx YZ Tc ~~~~~~~~~~~~~~~~~~~~~~~~~~~

YZ --> c d b/c/e
na --> e f
Xx --> c/d/e b/c/d/e
Tc --> b/d c

P(c|c) = 0.2                P(c|b) = 1.0
P(d|c) = 0.6                P(e|e) = 0.333
P(b|c) = 0.2                P(f|e) = 0.333
P(b|d) = 0.25               P(d|e) = 0.333
P(e|d) = 0.25               P(e|f) = 1.0
P(c|d) = 0.5

~~~~~ c c d b c d e e f e d c d c b c ~~~~~~~~~~~~~~~~~~~~~~~
```

Figure 1. Construction of a hierarchical Markov model.

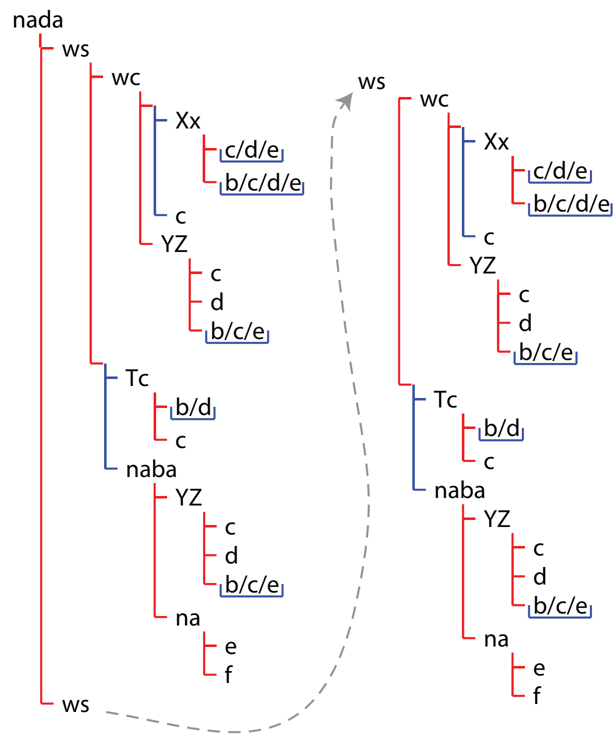

Figure 2. Tree structure of the model.

that of Table 1: hierarchical levels are arranged from bottom-to-top. At the bottom level, we see the original sequence. Immediately above that we see the derived statistical model for the sequence. In this case, it is a regular, first-order Markov model. Above that, we have the rewrite rules derived for the informationally optimal encoding. This encoding then becomes the source data for the next level and successive sections of the figure correspond to successive levels of the hierarchy in the obvious way. A graphical representation of the branching structure of the model is shown in Figure 2.

To use this hierarchical Markov model for generative purposes, we take the top-level symbol ('nada' in this case)[5] and recursively expand it using relevant rewrite rules. Whenever application of a rewrite rule offers a choice, selection is made using the Markov rule, i.e., the probability of a particular alternative being selected is the observed conditional probability of that alternative.

Formal definition of the method relies on

$$E(S) = \underset{R_S}{\operatorname{argmax}}\, \dot{G}(R_S) \tag{1}$$

This defines $E(S)$, the informationally optimal, rewrite-based encoding of S, to be the application (to S) of a rewrite-system R that maximizes information per symbol $\dot{G}$. On this basis, the modeling method can be defined as

$$\hat{M}(S) = \langle M(S), \hat{M}(E(S)) \rangle \tag{2}$$

The hierarchical Markov model $\hat{M}$ of any sequence S is defined to consist of a regular Markov model $M(S)$ and the hierarchical Markov model of S's optimal, rewrite-based encoding. If S contains a single symbol, $\hat{M}(S)$ is undefined.

5. EXPERIMENTS

Some experiments with musical sequences have been carried out. These have explored use of a variety of sources with the most interesting results being obtained with single-instrument (e.g., piano) compositions. Compositions from the Baroque period have proved particularly fruitful.

Data for the experiments were obtained by translating public-domain MIDI files into note-sequence form. This involved extracting note events and replacing absolute timing information (the MIDI 'tick' value) with relative values (i.e., timing offsets). This enabled implicit capture of

[5] All non-terminal symbols in this example are random labels.

rhythm while use of zero offsets was the means of capturing polyphony. Datasets from different genres were obtained and evaluated under various sampling regimes. Figure 3, for example, illustrates application of the approach to Bach's prelude No. 4 in C# Minor from Book 1 of the Well-Tempered Clavier. The upper part of the figure shows the midi data used, with the vertical axis representing midi pitch value (for a particular noteOn/noteOff combination) and the horizontal axis being midi tick. The lower part shows a hierarchical Markov reproduction. A MIDI version of this can be played at www.christhornton.eu/demos/reconstituted-music-examples.html.

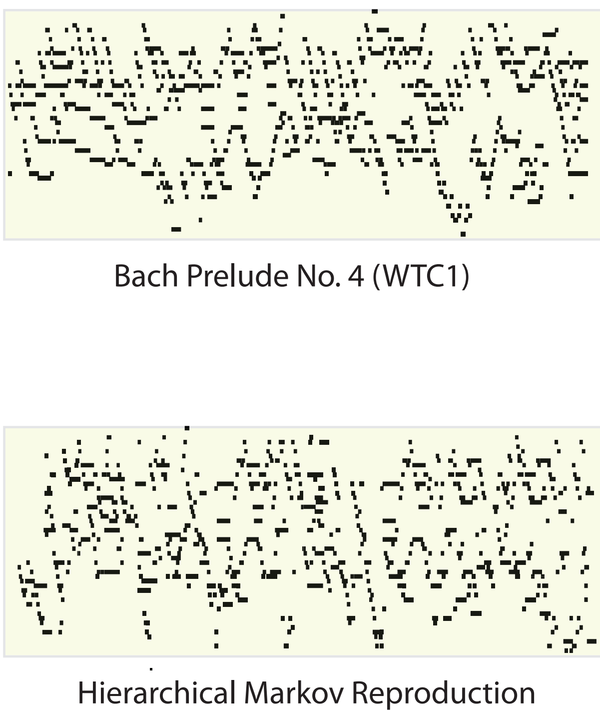

Figure 3. Hierarchical Markov reproduction of Bach's Prelude in C# Minor.

6. DISCUSSION

In any application involving modeling of music, it is possible to follow a broadly statistical approach or a broadly hierarchical one. Statistical approaches offer the benefit of simple sampling methods but may constrain the level of analysis. Hierarchical approaches offer flexibility in the level of analysis but may require use of special-purpose sampling and discovery, which may necessitate bringing domain-specific descriptors into play [10].

The proposed approach combines the sampling advantages of a statistical/informational approach with the flexibility of a hierarchical method. No special-purpose sampling is involved. As in hierarchical dictionary derivation [8], statistical functionality is adapted to the purpose of deriving a hierarchical model. This then forms the context for development of a nested structure of conventional statistical models. The benefit of this strategy is realized at the generative stage. Symbol expansion is modulated in a way that enforces structural organization at multiple levels of analysis.

7. REFERENCES

[1] Verbeurgt, K., Dinolfo, M. and Fayer, M. (2004). Extracting patterns in music for composition via markov chains. *IEA/AIE'2004: Proceedings of the 17th International Conference on Innovations in Applied Artificial Intelligence* (pp. 1123-1132). Springer Verlag.

[2] Paiement, J., Grandvalet, Y., Bengio, S. and Eck, D. (2007). Generative model for rhythms. *Proceedings of NIPS 2007* (Music, Brain and Cognition Workshop).

[3] Conklin, D. and Witten, I. (1995). Multiple viewpoint systems for music prediction. *Journal of New Music Research*, *24*, No. Issue 1 (pp. 51-73).

[4] Chai, W. and Vercoe, B. (2001). Folk music classification using hidden markov models. *Proceedings of the International Conference on Artificial Intelligence*. Las Vegas, Nevada.

[5] Scaringella, N., Zoia, G. and Mlynek, D. (2006). Automatic genre classification of music content: a survey. *Signal Processing Magazine,*, *23*, No. 2 (p. 133). IEEE.

[6] Dubnov, S., Assayag, G., Lartillot, O. and Bejerano, G. (2003). Using machine-learning methods for musical style modeling. *IEEE Computer*, *36*, No. 10 (pp. 73-80).

[7] Pearce, M., Mullensiefen, D. and Wiggins, G. (2008). *An Information-dynamic Model of Melodic Segmentation*. Helsinki, Finland: International Workshop on Music and Machine Learning.

[8] Witten, I. (2004). Adaptive text mining: inferring structure from sequences. *Journal of Discrete Algorithms*, *2*, No. 2 (pp. 137-159).

[9] Li, M. and Vitányi, P. (1997). *An Introduction to Kolmogorov Complexity and Its Applications: Second Edition*. New York: Springer-Verlag.

[10] Conklin, D. (2002). Representation and discovery of vertical patterns in music. In C. Anagnostopoulou, M. Ferrand and A. Smaill (Eds.), *Music and Artificial Intelligence: Proceedings ICMAI 2002* (pp. 32-42). Springer-Verlag,.