HMM Analysis of Musical Structure: Identification of Latent Variables Through Topology-Sensitive Model Selection

Panayotis Mavromatis

Department of Music and Performing Arts,
New York University,
35 West 4th St., Suite 777,
New York, NY 10012, USA
panos.mavromatis@nyu.edu
http://theory.smusic.nyu.edu/pm/

Abstract. Hiden Markov Models (HMMs) have been successfully employed in the exploration and modeling of musical structure, with applications in Music Information Retrieval. This paper focuses on an aspect of HMM training that remains relatively unexplored in musical applications, namely the determination of HMM topology. We demonstrate that this complex problem can be effectively addressed through search over model topology space, conducted by HMM state merging and/or splitting. Once successfully identified, the HMM topology that is optimal with respect to a given data set can help identify hidden (latent) variables that are important in shaping the data set's visible structure. These variables are identified by suitable interpretation of the HMM states for the selected topology. As an illustration, we present two case studies that successfully tackle two classic problems in music computation, namely (i) algorithmic statistical segmentation and (ii) meter induction from a sequence of durational patterns.

1 Introduction

Hiden Markov Models have been successfully employed in the exploration and modeling of musical structure [1,2], with applications in Music Information Retrieval [3].

Simply put, a Hidden Markov Model is a probabilistic version of a *Finite State Machine* (FSM), or formal specification of a finite state grammar. A FSM is formally defined by *states* and *transitions*, graphically represented by circles and arrows respectively. A FSM generates a symbolic sequence by traversing a path of states connected by transitions, following the direction of the arrows. the generated sequence is the string of output symbols encountered in the path. A FMS is a simple and flexible way to specify finite-memory constraints on the symbolic values of variables that characterize musical structure (e.g., pitch, duration, etc.) and as such offers useful formal characterizations of the structure of musical sequences.

E. Chew, A. Childs, and C.-H. Chuan (Eds.): MCM 2009, CCIS 38, pp. 205–217, 2009.

A *Hidden Markov Model* (HMM) is a FSM with probabilities attached to its transitions and output symbols [4,5]. The generation of a sequence through a specific HMM path has probability equal to the product of all transition and output probabilities encountered in traversing the generating path.

What gives the HMM technique its strength and flexibility is the fact that selecting the best HMM for a given data set can be generally accomplished through efficient algorithms. For instance, given a data set of symbolic sequences whose structure we wish to explore, it is customary to assume a HMM of fixed topology (i.e., number of states, and how they are connected by transitions) and identify the model parameters (i.e., transition and output probabilities) that best fit the data set, in the sense of Maximum Likelihood Estimation, using the so-called *Baum-Welch* algorithm.

This paper focuses on an aspect of HMM training that remains relatively unexplored in musical applications, namely the determination of HMM *topology.* Our aim is to algorithmically construct models whose topologies consist of states interpretable as values of latent ("hidden") variables that may play important role in the determination of musical structure. In a given application, one may wish to focus on a particular ("visible") musical variable, aiming to model syntactical constraints on its successive values (e.g., stylistically acceptable patterns of note durations). The states of a HMM obtained through topology-sensitive search should indicate which additional variables must be taken into consideration (e.g., metric position) in order to understand the syntax of the original "visible" variable that one set out to model. This can be accomplished by showing a close correspondence between HMM states and particular values of the candidate "hidden" variables.

For an HMM topology to be interpretable in the manner suggested in the preceding paragraph, special effort must be put in the topology selection algorithm. If one simply relies on Baum-Welch optimization of the HMM parameters, one will in most cases obtain HMMs whose states are not readily interpretable, however well these models may fit the data. Previous studies that attempted to address this complex problem have generally employed some form of search over model topology space, which was conducted by HMM state merging [6] or splitting [7]. In this paper, we use the same basic search procedure, except that we allow state merging and splitting to be combined in the same search. In addition, we evaluate each candidate model using a Bayesian approach, in which a HMM's prior probability is determined through the Minimum Description Length principle. This prior is optimal in that it leads to models that are neither too large nor too small, and has been found to provide a reliable termination criterion for the state merging/splitting search.

We will illustrate our method with the help of two case studies that successfully tackle two classic problems in music computation, namely (i) algorithmic statistical segmentation and (ii) meter induction from a sequence of durational patterns.

2 HMM Training and Topology Identification

The proposed method of topology identification takes place in the framework of Bayesian model selection. More specifically, given data set D, we seek the model M that maximizes the probability $P(M|D)$ of the model given the data. The latter is obtained through Bayes's Law as

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}$$

It is customary to use the simpler form

$$P(M|D) \propto P(D|M)P(M) \tag{1}$$

since $P(D)$ is constant over models M and therefore does not affect the maximization problem. $P(M)$ is known as the *model prior probability*, assigned to the model on general grounds before the data set is consulted. Likewise, $P(M|D)$ is known as the *model posterior probability*, and represents the probability of the model after the data has been taken into consideration.

Topology identification is achieved through a suitable choice of model prior $P(M)$, defined as a function of model topology alone, and designed to reward model simplicity. For a fixed topology, $P(M)$ is fixed, and so maximization of the model posterior amounts to maximizing the $P(D|M)$ part in eq. (1). This is achieved through the Baum-Welch (BW) algorithm, which chooses the model parameters maximizing the probability of the data set using the Expectation-Maximization principle. Overall, the maximization problem defined by eq. (1) is a concrete implementation of Occam's Razor, and achieves optimal balance between goodness-of-fit and model simplicity.

We have shown elsewhere [8] that an optimal choice for $P(M)$ is a *model complexity prior* given by

$$P(M) = Ke^{-D(M)} \tag{2}$$

where the function $D(M)$ is defined by

$$D(M) \equiv L(n_S) + L(d) + n_S \, log \frac{(d + n_S + 1)!}{d!n_S!} + n_T \, log \frac{(d + n_A + 1)!}{d!n_A!} \tag{3}$$

and $L(n)$ is the *universal prior for integers* [9, pp. 34–5], defined by

$$L(n-1) = c + log(n) + log(log(n)) + log(log(log(n))) + \ldots \tag{4}$$

Here n_S is the number of HMM states, n_A is the number of distinct output symbols in the data sequences, and K and c are suitably chosen normalization constants. An additional integer d represents the decimal precision needed to express the real-valued HMM model parameters. The expression in eq. (3) was derived in [8] with the help of the Minimum Description Length principle [9,10].

The best way to tackle the problem of HMM topology selection is by systematizing the search over all possible HMM graphs. Such a search scheme typically begins with an extreme graph which is maximally simple or maximally complex. Incremental improvements are subsequently performed on each candidate graph by either (i) *splitting* one of its states, if the graph is too simple, or (ii) *merging* two of its states if the graph is too complex. As an illustration, the following procedure formalizes the state-splitting search:

1. Begin with a *one-state* HMM. This model has only one transition, namely the one from the single state to itself. The output probabilities on that transition can be determined by the BW algorithm.
2. For this and each subsequent candidate model,
   (a) Choose a state to split. Determine the new graph that results from the splitting.
   (b) Perform BW estimation of the new graph's parameters.
   (c) Evaluate the resulting HMM's posterior probability using eq. (1) with the model complexity prior (eqs 2–4).

   Continue Steps (a–c) until all the states have been tried for splitting. The split-state HMM with the best posterior becomes the next candidate model, and Step 2 is repeated for as long as the candidate models' posterior probability continues to improve.
3. The process terminates once the posterior probability of the candidate model begins to deteriorate, and the HMM with the highest overall posterior is identified as the optimal HMM for the given data set.

One can modify Step 2(a) above to replace state-splitting by state-merging. Alternatively, one can consider both possibilities at each step, choosing the option that maximizes the model posterior at that step.

The above HMM topology selection process will now be illustrated with the help of two case studies.

3 Case Study I: Statistical Segmentation of Symbolic Sequences

Statistical segmentation is used to refer to the process of identifying grouping boundaries in sequences based solely on the patterns of occurrences of symbol combinations, without relying on explicit cues or annotations for such boundaries.

The process can be illustrated with the help of a data set $D1$ based on a language that was artificially synthesized to investigate statistical learning of tone sequences by people in an experimental setting [11]. The set of symbols, or *alphabet*, for this artificial language consists of pitches of the chromatic scale, to be represented by the symbols $\{\mathtt{C}, \mathtt{C}\sharp, \mathtt{D} \ldots \mathtt{B}\}$. The data sequences of $D1$ are built out of the following six three-symbol artificial segments ("words"):

A D B   D F E   G G♯ A   F C F♯   D♯ E D   C C♯ D

These words appear randomly with equal probability in the sequences of our data set $D1$. (Word combinations were more restricted in Saffran's stimuli, due

Table 1. Calculation of model posteriors for all the HMMs considered in the word segmentation example involving data set $D1$. Each model is obtained from the previous one by state splitting. The first column shows the HMM's number of states n_S. The second column shows the state split from which that model was obtained. Negative logarithms of probability values are used throughout. The selected model maximizes the model posterior or, equivalently, minimizes the value in Column 5. This model is marked with an asterisk in Column 1.

n_S	State Split	$-log_2 P(D\|M)$	$-log_2 P(M)$	$-log_2 P(M\|D)$
1	-	20365.8	84.2112	20450.1
2	0	15539.9	159.162	15699
3	0	13570.1	220.761	13790.9
4	1	12116.8	300.335	12417.2
5	3	11091.5	326.96	11418.4
6	2	10250.1	439.757	10689.8
7	0	9527.7	400.767	9928.47
8	1	8607.96	480.044	9088
9	4	7992.11	409.265	8401.38
10	7	7385.11	428.647	7813.76
11	6	6798.11	448.685	7246.8
12	5	6220.11	469.327	6689.44
13*	0	5653.91	624.18	6278.09
14	2	5653.91	670.633	6324.54

to the experimental design.) A typical sequence in $D1$ will therefore look like this:

G G♯ A A D B D♯ E D A D B C C♯ D D F E (5)

The output of the segmentation will be the same sequence annotated with word boundaries as follows:

G G♯ A / A D B / D♯ E D / A D B / C C♯ D / D F E

Our HMM analysis was applied to a data set $D1$ constructed in the above manner, consisting of 200 randomly generated sequences with an average length of 27.21 symbols. A state-splitting search was performed to identify the best HMM topology. Each candidate split was followed by Baum-Welch estimation of the HMM parameters. The results of this search are summarized in Table 1. The model identified as the winner is the one that carries the maximum posterior probability. This model is marked with an asterisk in the first column of the table. The model's graph structure is given in Figure 1.

To illustrate how the HMM of Figure 1 performs segmentation on a data sequence, it is helpful to consider the *most likely* HMM path that generates the sequence in question, also known as the sequence's *Viterbi* path [4,5, pp. 331–3]. For the sequence of example (5), this path turns out to be the following:

$$\begin{array}{cccccccccccccccccccc} & \mathit{BEGIN} & \mathtt{G} & & \mathtt{G\sharp} & & \mathtt{A} & & \mathtt{A} & & \mathtt{D} & & \mathtt{B} & & \mathtt{D\sharp} & & \mathtt{E} & & \mathtt{D} & \\ s_0 & \rightarrow & s_1 & \rightarrow & s_2 & \rightarrow & s_3 & \rightarrow & s_1 & \rightarrow & s_8 & \rightarrow & s_9 & \rightarrow & s_1 & \rightarrow & s_{12} & \rightarrow & s_4 & \rightarrow s_1 \end{array}$$

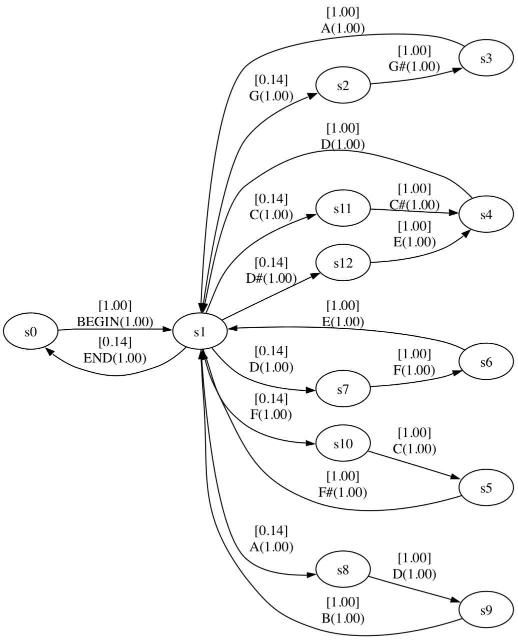


Fig. 1. The best HMM for data set $D1$, obtained through state-splitting

$$\begin{array}{ccccccccccccccccccccccc} & \mathtt{A} & & \mathtt{D} & & \mathtt{B} & & \mathtt{C} & & \mathtt{C\sharp} & & \mathtt{D} & & \mathtt{D} & & \mathtt{F} & & \mathtt{E} & & END & & \\ s_1 & \rightarrow & s_8 & \rightarrow & s_9 & \rightarrow & s_1 & \rightarrow & s_{11} & \rightarrow & s_4 & \rightarrow & s_1 & \rightarrow & s_7 & \rightarrow & s_6 & \rightarrow & s_1 & \rightarrow & s_0 \end{array} \tag{6}$$

With the help of this Viterbi path, all word boundaries in the sequence are clearly identified through the HMM state s_1. The significance of that state as a marker of word boundaries can also be confirmed by observing the graph structure of Figure 1 and following the derivation path of any sequence generated by that graph.

This simple example serves to illustrate that, just like the experimental subjects in the study by Saffran et al. [11], the HMM topology selection technique presented here can exploit the statistical structure of symbolic sequences to segment them into grouping units. This result is replicated with other similar data sets and suggests that—at least in certain cases—segmentation can be performed on the basis of statistical information alone, without recourse to other structure, such as Gestalt principles of grouping.

4 Case Study II: Meter Induction from Rhythmic Patterns

Meter induction refers to the inference of metrical structure from a pattern of note durations. Our second case study illustrates this process by analyzing patterns of durations found in Palestrina's vocal music. Table 2 lists all the possible note and rest durations employed in the style.

It should be noted that the goal of this application is not to do meter induction *per se*. Rather, we seek to model Renaissance rhythm by establishing a syntax of note durations. With the help of the HMM topology selection technique, we hope to identify any other variable(s) that may be most relevant in constraining and shaping the style's duration patterns. In this instance, the most crucial variable turns out to be metric placement, and is identified by the interpretation of HMM states as explained below.

The HMM analysis of the present case study was performed on a sample of melodies taken from the corpus of Palestrina's masses. The corpus was obtained from the Internet in *Humdrum*-encoded form.[1] The sample was constructed as follows:

1. The corpus of Palestrina masses was subdivided into movements, or sections of movements. Each such section was further subdivided into individual vocal lines. This processing was carried out using standard Humdrum tools. The result was a database of 5034 vocal lines covering the entire corpus.
2. Out of these 5034 vocal lines, fifty were chosen at random to form the sample, using a random number generator.
3. Each of the fifty lines was further subdivided into one or more data sequences. The divisions were made at places where there was a rest of one complete bar or longer. This subdivision was intended to ensure that the data sequences represented units close to the phrase level.
4. Finally, the durations of each data sequence were extracted and encoded using the symbols listed in the fourth column of Table 2.

An example of this encoding is shown above the staff in Figure 3. The resulting sample consisted of 190 such sequences with an average length of 34.48 symbols.

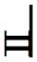
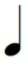
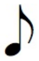
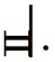
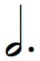
The results of HMM inference algorithm are shown on Table 3. The HMM with the highest posterior probability was a 6-state model, marked with an asterisk in the first column of the table. Figure 2 shows the model in graph form.

As in the previous case study, the model's structure will be easier to interpret with the help of the data sequences' Viterbi path. As an illustration, the Viterbi path for a typical melody in the data set is given in Figure 3.

Examination of the state sequences in the model's Viterbi paths reveals one striking property: there is a close correspondence between the HMM states and the various metric positions in the compositions' underlying 4/2 meter. As can be seen from the example of Figure 3, states s_1 and s_3 occur exclusively on strong beats (1 or 3), whereas state s_2 only occurs on weak beats (2 or 4); moreover, state s_4 only occurs on weak quarters, and the rare occurrence of state s_5 coincides with a weak eighth-note subdivision. In other words, the HMM appears to be "aware" of metric placement for each duration it generates. This

[1] URL: *http://csml.som.ohio-state.edu/HumdrumDatabases/classical/Renaissance/Palestrina/Masses/* (last visited March 2009).

Table 2. The note and rest durations available to the Renaissance vocal style. These are shown along with the corresponding symbolic value of the duration variable, as encoded for the HMM analysis of the present project. The rightmost column records the possible metric placements for each duration, as prescribed in counterpoint instruction.

Music symbol	Renaissance name	Modern name	Encoding	Metric position
	Longa		`L`	beats 1, 3
	Breve		`B`	beats 1, 3
	Semibreve	Whole note	`W`	beats 1, 2, 3, 4
	Minim	Half note	`H`	beats 1, 2, 3, 4
	Semiminim	Quarter note	`Q`	any quarter
	Fusa	Eighth note	`E`	pairs, weak quarter
	Dotted Longa		`L.`	beats 1, 3
	Dotted Breve		`B.`	beats 1, 3
	Dotted Semibreve	Dotted whole note	`W.`	beats 1, 3
	Dotted Minim	Dotted half note	`H.`	beats 1, 2, 3, 4
	Semibreve rest	Whole note rest	`Rw`	beats 1, 3
	Minim rest	Half note rest	`Rh`	beats 1, 3

awareness is embodied in the HMM states, whose job is to encapsulate the most decisive factors that determine the next output at each point in time. The fact that each HMM state has chosen to incorporate metric information should perhaps come as no surprise, given the generally acknowledged role of metric constraints in the style's rhythmic syntax. What is perhaps most remarkable is that metric position was not originally encoded explicitly in the data sequences. The HMM inference algorithm was able to detect the importance of this variable, based on statistical regularities in the sequential combinations of note durations.

Table 3. Calculation of model posteriors for all the HMMs considered in the analysis of Palestrina rhythm. As in the earlier example, each model is obtained from the previous one by state splitting. The columns of this table carry the same interpretation as those of Table **??**.

n_S	State Split	$-log_2P(D\|M)$	$-log_2P(M)$	$-log_2P(M\|D)$
3	-	14091.2	826.489	14917.7
3	0	14091.2	826.489	14917.7
4	1	12950.7	942.963	13893.7
5	0	12161.9	1051.110	13213.0
6*	4	12002.0	1204.640	13206.6
7	0	12002.0	1363.340	13365.3

Examination of the HMM states reveals a close correspondence between HMM states and the rules of metric placement found in standard Renaissance counterpoint textbooks [12,13], including the constraints on each duration's metric placement, and the general tendency to find longer note values near the beginnings and ends of phrases. The latter property is reflected in the differentiation between the two "strong beat" states s_1 and s_3; the former represents strong beats near the beginning and end of phrases, whereas the latter occurs in the phrases' interior positions.

5 Conclusions

The two case studies presented in this paper have demonstrated how topology-sensitive HMM training can successfully uncover hidden structure underlying the observable behavior of symbolic data sequences. Indeed, generic application of the Baum-Welch algorithm would not have resulted in readily interpretable graphs such as those of Figures 1 and 2. Only when HMM training incorporates model topology identification, in a way that is sensitive to the data set's statistical regularities, will the HMM states be readily interpretable in terms of the processes underlying the data sequence's generation. In such cases, we can interpret the different HMM states as representing the values of *hidden*, or *latent*, variables that are most crucial in shaping the structural constraints of the data sequences.

More specifically, one salient latent variable underlying Case Study I could be identified as "word completion status" with the two values 'yes' (corresponding to state s_2) and 'no' (corresponding to states s_1 s_3, and s_4); furthermore, a second latent variable of "word label" could account for the differences among the non-boundary states s_1 s_3, and s_4. For Case Study II, the most salient latent variable seemed to be "metric position" with most HMM states representing distinct values. A second latent variable representing "position in the phrase" was found to differentiate between states s_1 and s_3.

Of course, in both the above examples, identification of the relevant latent variables is relatively straightforward. This is because the HMM graphs are rather small, and so the correspondence between HMM states and latent variable values can be directly perceived. In more complicated situations, however, this need not be the case. We must have a way of interpreting HMM states that is more reliable than simple inspection. In general, the interpretation process could be systematized by compiling contingency tables that show how each HMM state aligns, or doesn't align, with the values of a set of candidate latent variables along the HMM paths that generate the data set (the Viterbi path offering the dominant contribution).

Finally, it should be noted that, as our experiments with various data sets indicate, our MDL prior of eq. (2–4) is an essential ingredient for the identification of the right model topology. Other priors that we have tried typically produce smaller graphs—e.g., caused by premature termination of state-splitting—whose states are not consistently interpretable. In general, whenever the data is abundant, it is found that the result is less sensitive to the choice of prior. However, that choice really matters when data is scarce, which is the case, for example, in historically delimited musical corpora (e.g. "all D-mode Gregorian tracts"). The MDL approach is a strongly motivated and principled way of choosing a prior, which in the majority of cases leads the topology search to discover interpretable graphs.

It should be also noted that a simple splitting/merging search over model topologies, unaided by other search heuristics, does not always yield readily interpretable graphs, especially in data sequences with rich alphabets of symbols. The problem is that the splitting/merging search is a form of "best first" search that guarantees an optimal next step in the search, leading to a local maximum of the model posterior; however, it cannot guarantee that the maximum reached in this way will be optimal in the global sense. This is of course a concern for any optimization problem. We have found that, in order to produce interpretable results in the most general cases, the search proposed in this paper has to be augmented with heuristics that determine an appropriate starting point for the splitting or merging. This issue is currently under investigation, and will be presented in a future work.

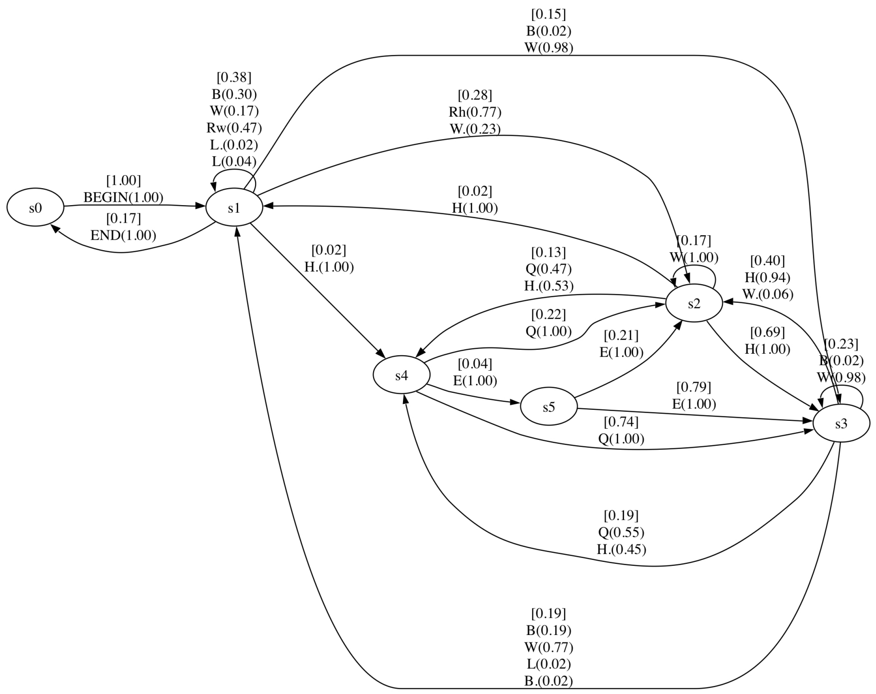


Fig. 2. The best HMM for the data sequences of durations in the Palestrina sample. The graph's output symbols are encoded using the symbols listed in the fourth column of Table 2.

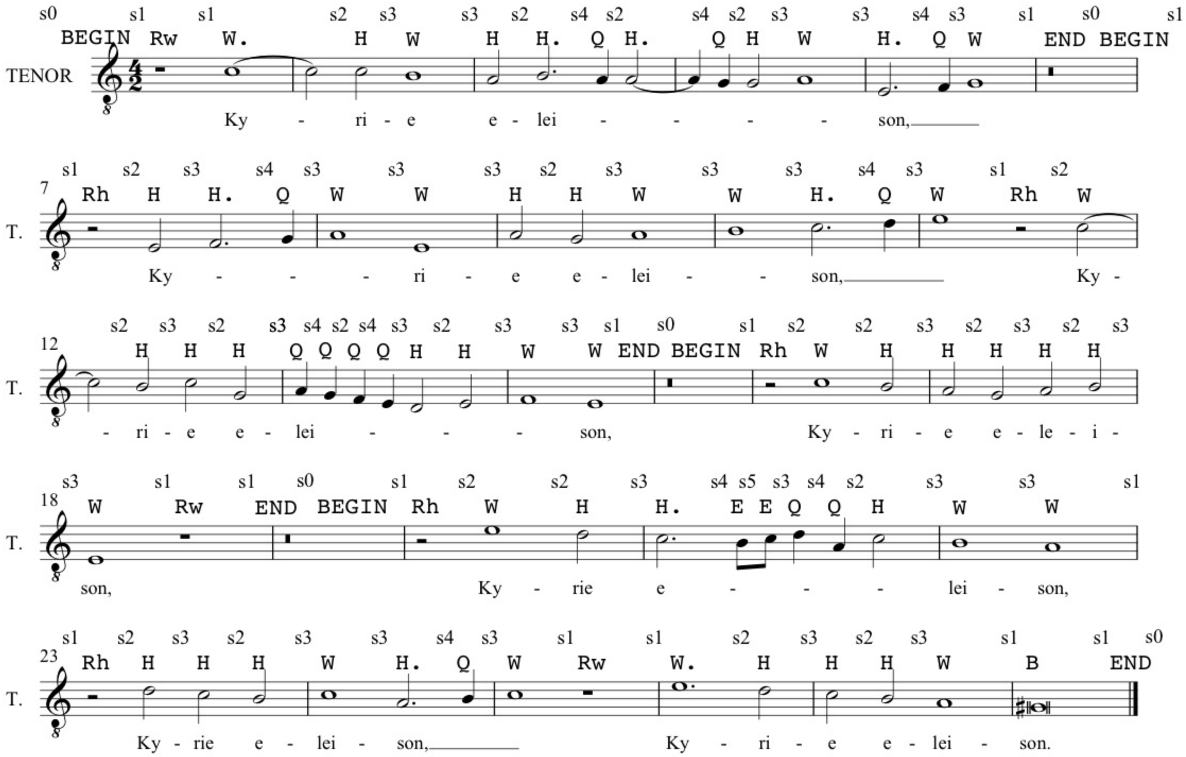

s0 s1 s1 s2 s3 s3 s2 s4 s2 s4 s2 s3 s3 s4 s3 s1 s0 s1
BEGIN Rw W. H W H H. Q H. Q H W H. Q W END BEGIN
TENOR
Ky - ri - e e - lei - - - - son,
s1 s2 s3 s4 s3 s3 s3 s2 s3 s3 s3 s4 s3 s1 s2
7
Rh H H. Q W W H H W W H. Q W Rh W
T.
Ky - - - ri - e e - lei - - son, Ky -
s2 s3 s2 s3 s4 s2 s4 s3 s2 s3 s3 s1 s0 s1 s2 s2 s3 s2 s3 s2 s3
12
H H H Q Q Q Q H H W W END BEGIN Rh W H H H H H
T.
- ri - e e - lei - - - son, Ky - ri - e e - le - i -
s3 s1 s1 s0 s1 s2 s2 s3 s4 s5 s3 s4 s2 s3 s3 s1
18
W Rw END BEGIN Rh W H H. E E Q Q H W W
T.
son, Ky - rie e - - - lei - son,
s1 s2 s3 s2 s3 s3 s4 s3 s1 s1 s2 s3 s2 s3 s1 s1 s0
23
Rh H H H W H. Q W Rw W. H H H W B END
T.
Ky - rie e - lei - son, Ky - ri - e e - lei - son.

Fig. 3. A typical Palestrina vocal line (from *Missa Te Deum Laudamus*, *Kyrie* II, Tenor I) annotated with its Viterbi path. The state sequence corresponding to that path is marked with the symbols $s_0 - s_5$ above each staff. Arrows from one state to the next have been suppressed for visual clarity. The output symbols corresponding to the encoded durations appear below the state sequence and above the corresponding note or rest. Duration encodings are listed in column 4 of Table 2.

References

1. Raphael, C., Stoddard, J.: Functional Harmonic Analysis Using Probabilistic Models. Computer Music Journal 28, 45–52 (2004)
2. Mavromatis, P.: A Hidden Markov Model of Melody Production in Greek Church Chant. Computing in Musicology 14, 93–112
3. Bello, J.P., Pickens, J.: A Robust Mid-Level Representation for Harmonic Content in Music Signals. In: Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005), London, UK (September 2005)
4. Rabiner, L.R., Juang, B.-H.: An Introduction to Hidden Markov Models. IEEE ASSP Magazine 3, 4–16 (1986)
5. Manning, C.D., Schütze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)
6. Stolcke, A., Omohundro, S.M.: Hidden Markov Model Induction by Bayesian Model Merging. In: Hanson, S.J., Cowan, J.D., Giles, C.L. (eds.) Advances in Neural Information Processing Systems, vol. 5, pp. 11–18. Morgan Kaufmann, San Mateo (1993)
7. Ostendorf, M., Singer, H.: HMM Topology Design Using Maximum Likelihood Successive State Splitting. Computer Speech and Language 11, 17–41 (1997)
8. Mavromatis, P.: Minimum Description Length Modeling of Musical Structure. The Journal of Mathematics and Music (under revision) (submitted to)
9. Rissanen, J.: Stochastic Complexity in Statistical Inquiry. Series in Computer Science, vol. 15. World Scientific, Singapore (1989)
10. Grünwald, P.D.: The Minimum Description Length Principle. Adaptive Computation and Machine Learning. MIT Press, Cambridge (2007)
11. Saffran, J.R., Johnson, E.K., Aslin, R.N., Newport, E.L.: Statistical Learning of Tone Sequences by Human Infants and Adults. Cognition 70, 27–52 (1999)
12. Jeppesen, K.: Counterpoint: The Polyphonic Vocal Style of the Sixteenth Century. Prentice Hall, New York (1939); reprinted by Dover (1992)
13. Gauldin, R.: A Practical Approach to Sixteenth-Century Counterpoint. Waveland Press, Long Grove (1995)