Chapter 3
Markov Models

Markov models were introduced by the Russian mathematician Andrey Andreyevich Markov (1856–1922). Markov was a student of Pafnuty Chebyshev and worked amongst others in the field of number theory, analysis, and probability theory. From 1906 on, Markov published his first works on time dependent random variables.

Fig. 3.1 Andrey Andreyevich Markov in 1886. Contemporary engraving.

In the context of urn experiments, observations on time dependent random variables were already made by Pierre Simon de Laplace and Daniel Bernoulli [2, p. 10ff]. The first application of this method was an extensive text analysis by Markov: "In 1913 [. . .] Markov had the third edition of his textbook[1] published. [. . .] In that edition he writes, 'Let us finish the article and the whole book with a good example of dependent trials, which can be regarded approximately as a simple chain'. In what has now become the famous first application of Markov chains, A. A. Markov studied the sequence of 20,000 letters in A. S. Pushkin's poem 'Eugeny Onegin', discovering that the stationary vowel probability is $p = 0.432$, that the probability of a vowel following a vowel is $p1 = 0.128$, and that the probability of a vowel following a consonant is $p2 = 0.663$ [. . .]." [2, p. 16]. The term "Markov chain" (in

[1] Basharin et al. are referring to Markov's paper "Ischislenie veroyatnostej."

this work, also referred to as “Markov model”) for this class of stochastic procedures was first used in 1926 in a publication of the Russian mathematician Sergey Natanovich Bernstein.

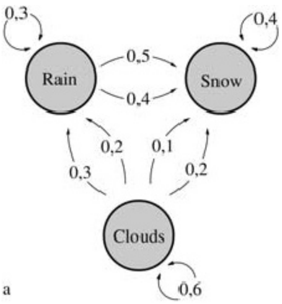
3.1 Theoretical Basis

The field of stochastics comprises probability calculus and statistics. *Stochastic processes* are used to describe a sequence of random events dependent on the time parameter (t). The set of events is called “*state space*,” while the set of parameters is known as the “*parameter space*.” If a stochastic process consists of a countable number of states, then it may also be referred to as a stochastic chain. In a stochastic chain, every discrete time t has a random variable X. In a *Markov chain*, it being a special kind of stochastic chain, the probability of the future state X_{t+1} (the random variable X at the time $t+1$) depends on the current state X_t. For the given times t_m and t_{m+1}, this probability is:

$$P(X_{tm+1} = j \mid X_{tm} = i) = p_{ij}(t_m, t_{m+1})$$

This expression indicates the *transition probability* of the state $X_{tm} = i$ at a given time t_m to the state $X_{tm+1} = j$ [3, p. 768].

A Markov chain can be represented by a state *transition graph*, or by a *transition matrix*, as can be seen in the example of a weather forecast in figure 3.2. The description of the edges in figure 3.2a shows the transition probabilities that can be found in transition matrix P represented as a table (3.2b). The sum of all transition probabilities in each state must equal 1. Starting from a particular state, the probabilities for a future state can be determined. These probabilities are calculated with the formula $p(t+k) = p(t) * P^k$; $p(t)$ representing the initial state, k the number of state transitions and P the transition matrix [3, p. 788].



a

Tomorrow`s Weather

Today`s Weather	Rain	Snow	Clouds
Rain	0,3	0,5	0,2
Snow	0,4	0,4	0,2
Clouds	0,3	0,1	0,6

b

Fig. 3.2 Representations of a first-order Markov chain.

If more than one past event is used in the calculation of the transition probabilities, then this is called a *higher-order Markov process*, the order indicating the number of past events that are relevant to the transition probabilities. Consequently, in a Markov model based on the tone pitches of a melodic corpus, the output sequence will more and more approach the structure of the corpus with increasing Markov orders.

If a particular sequence of length n does not occur in the corpus, then this sequence will not appear in the produced musical material in the corresponding Markov analysis and generation of *n*th order either. A possible solution to this problem is offered by so called "*smoothed n-grams*"[2] that use lower-order transition probabilities for the generation of higher-order transition probabilities. In this procedure, the missing transition probabilities for insufficient sequences of *n*th order can be acquired by interpolation with lower orders $n-1$, $n-2$, etc. At a given symbol at the current point, the transition probabilities calculated from the corpus get different weights for preceding sequences of different length. In general, Markov models can only occupy a finite number of states and can therefore be represented by finite automata[3] and within a graph representation. This possibility is especially interesting in cases where not all fields of the transition matrix need to be occupied [5, 11].

3.2 Hidden Markov Models

In hidden Narkov models (HMM), the sequences of the *observable output* symbols of a Markov model are visible, but their internal states and state transitions are not. In this case, the states of the "hidden" Markov models produce so-called *emission probabilities*, generating the musical segments (usually notes with tone pitches and lengths) as observable outputs. In the following, an HMM is explained through the example of a weather forecast. A news agency receives its information on political events in a foreign country from a correspondent at different times of the day. In his work, this correspondent is, amongst other things, influenced by the weather situation. So, for example, if the sun is shining, he likes to get up early and therefore sends his report already before breakfast. If it is raining, on the other hand, he likes to sleep a bit longer and accordingly does not start his daily routine before he has had some cups of strong tea. Consequently, depending on the time the report arrives, the news agency can also make inferences about the weather situation in the foreign country. But, because this is not the only determining factor for the correspondent's working discipline, the time the reports arrive may only suggest a particular weather situation. So, in analogy to an HMM, the specific times of arrival of the reports can be seen as the sequences of the observable output symbols of the HMM generated by the emission probabilities of the hidden states – the underlying probable sequence of different weather situations.

[2] See also [14, p. 6f].

[3] See chapter 4, Type-3 Grammar, DFA or NFA.

Hidden Markov models can deliver continuous as well as discrete distributions of emission probabilities. In algorithmic composition, however, continuous models are of little importance as the observed emissions are in most cases note values with quantized parameters. A hidden Markov model represents a coupled stochastic process, due to the transition probabilities of the states in the Markov model and the state-dependent emission probabilities of the observed events. The following indications and symbols are used for the formal description of an HMM [12, p. 7ff]:

N	Number of states in a Markov model
$\{S_1,\ldots,S_N\}$	Set of these states
$\pi = \{\pi_l,\ldots,\pi_N\}$	Vector of *initial probabilities* for each state (*initial state distribution*)
$A = \{a_{ij}\}$	Transition probabilities in the MM from state to state
M	Number of observable output symbols of the HMM
$\{v_1,\ldots,v_M\}$	Set of output symbols
$B = \{b_{jm}\}$	Emission probabilities as probability of output of a symbol in a state
T	Length of the output sequence
$O = O_1 \ldots O_T$	Sequence of output symbols with $O_t \in \{v_1,\ldots,v_M\}$
$Q = q_1 \ldots q_T$	Sequence of state sequences in the MM at output O with $q_t \in \{1,\ldots T\}$

A hidden Markov process with the three states S_1, S_2, S_3 and the three output symbols V_1, V_2, V_3 can be graphically represented as shown in figure 3.3.

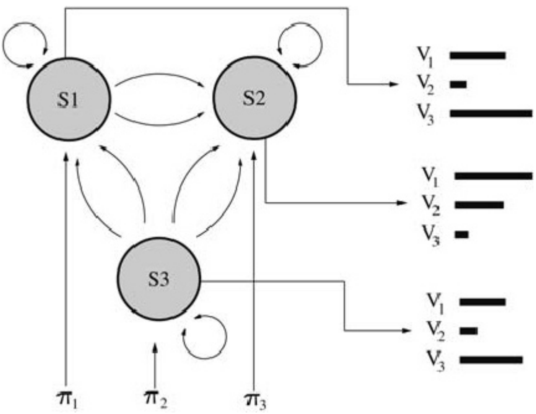


Fig. 3.3 Representation of a hidden Markov model.

In order to answer essential questions within a hidden Markov model, mostly three algorithms are applied:

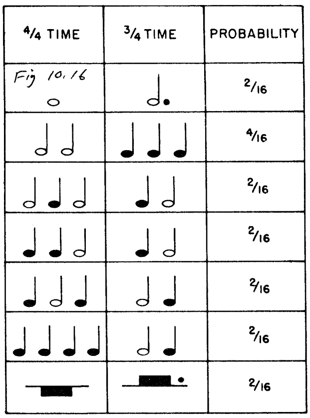
- The *forward algorithm* computes the probabilities for the occurrence of a particular observable sequence, where the parameters (transition and observation probabilities as well as the initial state distribution) of the HMM are known.

- The *Viterbi algorithm* calculates the most likely sequence of hidden states, called the *Viterbi path*, on the basis of a given observable sequence.
- The *Baum–Welch algorithm* is applied to find the most likely parameters of an HMM on the basis of a given observable sequence.

3.3 Markov Models in Algorithmic Composition

In algorithmic composition, the transition probabilities of a Markov model are generated either according to individual structural parameters, or calculated in the process of generating style imitations by analyzing a corpus.

Application of Markov processes in musical structure generation was first examined by Harry F. Olson (1901–1982) around 1950. Olson was an American electrical engineer and physicist who focused on acoustic research. Together with Henry Belar, he developed the "Electronic Music Synthesizer" in 1955, the first machine to be called a synthesizer. Olson analyzed eleven melodies by Stephen Foster and produced Markov models of first and second order in regard to pitches and rhythm (indicated by Olson as dinote- and trinote probabilities) [13, p. 430 ff]. Figure 3.4 shows results of the statistical analysis in regard to the probabilities of rhythmic patterns in 4/4 and 3/4 time as well as the occurence of certain pitch classes. Figure 3.5 shows a transition table for pitch classes corresponding to a first order markov process. For the standardization of the analysis all songs were transposed to the key of D.



Note	B_3	$C^{\#}_4$	D_4	E_4	$F^{\#}_4$	G_4	$G^{\#}_4$	A_4	B_4	$C^{\#}_5$	D_5	E_5
Relative frequency	17	18	58	26	38	23	17	67	42	29	30	17

Fig. 3.4 Probability of rhythms (top) [13, p. 433] and relative frequency of the notes (bottom) [13, p. 431] in eleven Stephen Foster songs. Reproduced with kind permission by Dover Publications.

Probability of following note

Note	B_3	$C^{\#}_4$	D_4	E_4	$F^{\#}_4$	G_4	$G^{\#}_4$	A_4	B_4	$C^{\#}_5$	D_5	E_5
B_3			16									
$C^{\#}_4$			16									
D_4	1	1	2	5	3	1		1		1	1	
E_4		1	6	3	4			1			1	
$F^{\#}_4$			2	4	5	2		2	1			
G_4					4	3		6	3			
$G^{\#}_4$								16				
A_4			1		5	1	1	4	3		1	
B_4			1		1	1		9	2		2	
$C^{\#}_5$									8		8	
D_5								4	7	3	1	1
E_5								6		10		

Probability of note following the preceding note expressed in sixteenths.

Fig. 3.5 Two-note sequences of eleven Stephen Foster songs [13, p. 431]. Reproduced with kind permission by Dover Publications.

The difference between these two models can be seen in the fact that the productions of 2nd order Markov models show more harmonious melody creations as well as better results for the end part of the composition.

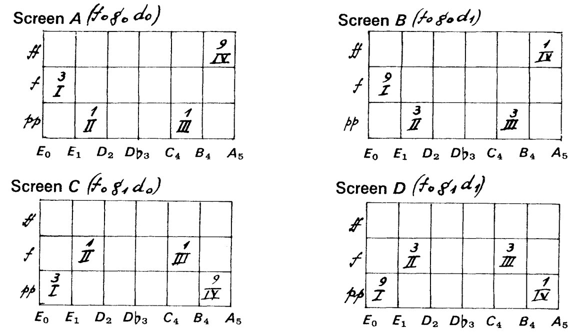
From 1955 on, Lejaren Hiller and Leonard Isaacson worked with the ILLIAC computer at the University of Illinois on a composition for string quartet: The "Illiac Suite" was performed for the first time in August 1956 and became famous as the first computer-generated composition.[4] Each of the movements, so-called "experiments," was dedicated to the realization of a special musical concept. In "experiment four," Hiller and Isaacson use Markov models of variable order for the generation of musical structure. Amongst others, these Markov models serve to select notes under various musical aspects, like the succession of skips and stepwise motions, the progression from consonant to dissonant intervals or even sound textures, which can be related to a tonal center in order to establish a distinct tonality.

Iannis Xenakis began to use Markov models for the generation of musical material[5] in 1958. In "Analogique A," Markov models are employed to arrange segments of differing density. Each of these segments, called "screens," consists of sounds of different dynamics. Figure 3.6 (top) shows four "screens" in which the lines represent a dynamic level and the columns a group of instruments. The transition probabilities of the "screens" result from a probability matrix for a 1st order Markov process as shown in figure 3.6 (bottom).

F.P. Brooks, A.L. Hopkins, P.G. Neumann and W.V. Wright [4] used a corpus of thirty-seven chorale melodies of similar metric-rhythmic structure for the Markov analysis and the subsequent generation. All chorales that are used for the generation model are in 4/4 time, begin on the last beat of the four beat measure, and do not contain note values shorter than an eighth note, which serves also as the basic rhythmic unit for the representation. Pitches within a range of four octaves are denoted by integers between 2 and 99, where all even numbers stand for a new note

[4] Cf. [8, p. 12], for the "Illiac Suite," also see chapters 2 and 10.

[5] Here: "Analogique A," "Analogique B" (1958) and "Syrmos" (1959).



↓	A $(f_0g_0d_0)$	B $(f_0g_0d_1)$	C $(f_0g_1d_0)$	D $(f_0g_1d_1)$	E $(f_1g_0d_0)$	F $(f_1g_0d_1)$	G $(f_1g_1d_0)$	H $(f_1g_1d_1)$
$A(f_0g_0d_0)$	0.021	0.357	0.084	0.189	0.165	0.204	0.408	0.096
$B(f_0g_0d_1)$	0.084	0.089	0.076	0.126	0.150	0.136	0.072	0.144
$C(f_0g_1d_0)$	0.084	0.323	0.021	0.126	0.150	0.036	0.272	0.144
$D(f_0g_1d_1)$	0.336	0.081	0.019	0.084	0.135	0.024	0.048	0.216
$E(f_1g_0d_0)$	0.019	0.063	0.336	0.171	0.110	0.306	0.102	0.064
$F(f_1g_0d_1)$	0.076	0.016	0.304	0.114	0.100	0.204	0.018	0.096
$G(f_1g_1d_0)$	0.076	0.057	0.084	0.114	0.100	0.054	0.068	0.096
$H(f_1g_1d_1)$	0.304	0.014	0.076	0.076	0.090	0.036	0.012	0.144

Fig. 3.6 Screens A-D and probability matrix in "Analogique A" [16, p. 89, 101]. Reproduced with kind permission by Pendragon Press.

event and the uneven ones represent a note held over from the previous one. For the analysis and further generation, the different chorale melodies are transposed to C major. Markov analyses up to the 8th order are generated for the whole corpus, as shown in figure 3.7a, for example, the representations called "octograms" can be seen, for each eighth element with seven same predecessors (character strings of length n are generally referred to as "n-grams"); the following columns contain amongst others the relative frequency of each eighth element. According to the mapping, e.g. the number sequence of the first group (* 36 37 26 27 32 33 26 22) represents the tone pitches $G\overline{G}C\overline{C}A\overline{A}CD$, the notes under the bar indicating a tie. With a basis of sequences of 64 segments with a duration of each 1/8 per chorale melody, the result is a total of 2368 constellations for 37 chorales, of which only 1701 are different. In figure 3.7b, top, the number of the different n-grams for the starting notes $G\overline{G}F\overline{F}E\overline{E}D\overline{D}C\overline{C}$ are shown (in the figure, Brooks et al. use m instead of n). In figure 3.7b, bottom, all different n-grams with struck notes, held notes or rests as starting values are listed.[6]

In the generation of musical material, some other conditions are also established, e.g. on particular metric stresses no pauses are allowed, or melodies have to end at particular scale degrees [4, p. 33ff]. Figure 3.8 shows some examples of musical

[6] Brooks et al. do not explain the meaning of the hyphen.

Cell 1	2	3	4	5	6	7	8	Octo-gram Count	Hepta-gram Count	Relative Fre-quency	Cumula-tive Probability
36	37	26	27	32	33	26	22	1			
*36	37	26	27	32	33	26	22	2		2/8	2/8
36	37	26	27	32	33	26	27	1			
36	37	26	27	32	33	26	27	2			
36	37	26	27	32	33	26	27	3			
36	37	26	27	32	33	26	27	4			
*36	37	26	27	32	33	26	27	5		5/8	7/8
*36	37	26	27	32	33	26	28	1	8	1/8	8/8
36	37	26	27	32	33	32	33	1			
*36	37	26	27	32	33	32	33	2	2	2/2	2/2
*36	37	32	33	32	33	32	33	1	1	1/1	1/1
36	37	32	33	32	33	36	37	1			
*36	37	32	33	32	33	36	37	2		2/3	2/3
*36	37	32	33	32	33	36	42	1	3	1/3	3/3

a

Initial Note	Order of Analysis m: 1	2	3	4	5	6	7	8
12 G	1	3	15	25	57 -	66	99 -	102
13 $\overline{G}$	1	12	30	67 -	84	124 -	131	150
16 F	1	6	14	25	43 -	45	63 -	63
17 $\overline{F}$	1	8	14	32 -	33	51 -	51	66
18 E	1	6	19	32	74 -	83	131 -	131
19 $\overline{E}$	1	11	23	68 -	78	130 -	135	156
22 D	1	4	19	30	66 -	69	104 -	109
23 $\overline{D}$	1	12	21	57 -	64	102 -	106	126
26 C	1	6	18	28	65 -	71	111 -	112
27 $\overline{C}$	1	13	25	65 -	72	112 -	113	136
All Struck Notes	18	47	152	219	444 -	479	698 -	705
All Held Notes	18	110	182	428 -	485	717 -	738	869
00 Initial Rest	1	5	10	28	45	70	95	127
Total Distinct m-Grams	37	162	344	675	974	1266	1531	1701

b

Fig. 3.7 Markov analyses and results in Brooks et al. [4, p. 32]. © 1993 Massachusetts Institute of Technology. By permission of The MIT Press.

production from 1st to 8th order. Furthermore, this study points out three problems that may occur in connection with structures generated by Markov models: Besides the danger of noticeable randomness of an output when applying a lower-order Markov model and overly restricted choices with higher orders, Brooks et al. also indicate a problem that may be easily overseen, which deals with the training corpus of the MM on whose basis the transition probabilities are determined. If the training data have a very similar structure, analyses of higher order may possibly be unnecessary, since, for instance, for each different transition probability of nth order only one possible transition of the order n + 1 may result. Therefore, an analysis of higher-order models does not contain any relevant information [4, p. 29].

Fig. 3.8 Production examples of different orders (m) by Brooks et al. [4, p. 38]. © 1993 Massachusetts Institute of Technology. By permission of The MIT Press.

3.3.1 Alternative Formalisms

An approach based on a production without previous analysis is described by Kevin Jones [11]. The author considers different relations that states of a MM may be in and establishes, based on the existent transition probabilities, "equivalence classes" of states within the Markov model. According to Jones, one class forms an accumulation of elements that are networked to a large extent, meaning that in their representation in a graph they can reach each other over the edges. Transitions between different classes are limited to only a few edges. Moreover, Jones distinguishes between "transient" and "recurrent classes" that are characterized by the fact that through the transition probabilities, states in "transient classes" may lead to states in other classes including "recurrent classes." If such a "recurrent class" is reached from "outside," it cannot be left any more, because the states within this class can only lead to states within the same class. Principally, this classification is a representation of an incomplete Markov model[7] where transition probabilities $p = 0$ are not taken into consideration and states which are related by transition possibilities $p > 0$ are interconnected by edges within a graph representation (figure 3.9).

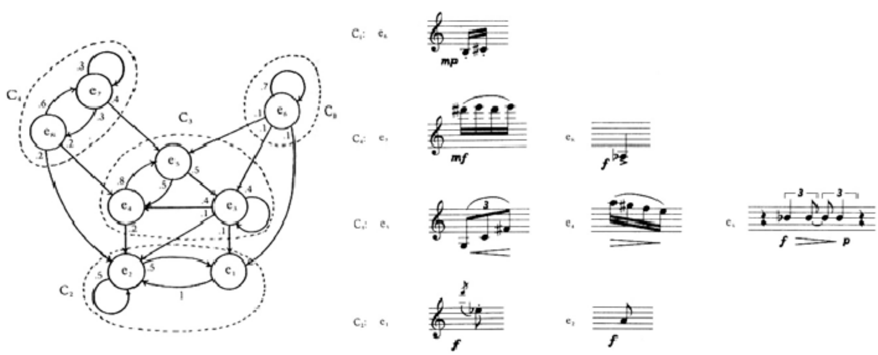


Fig. 3.9 Markov models in different equivalence classes (left), and terminals (right) [11, p. 384-385]. © 1993 by the Massachusetts Institute of Technology.

Jones's approach shows that for the states of a Markov model in the context of algorithmic composition, motivic structures instead of notes can also be used. Figure 3.10 demonstrates an example production of this model.

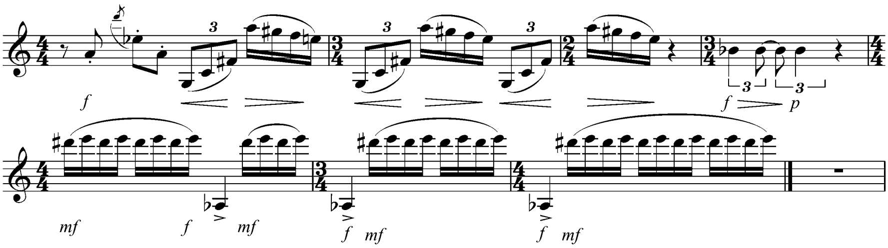


Fig. 3.10 Example production by the system of Jones.

In the comprehensive approach by Dan Ponsford, Geraint Wiggins and Chris Mellish [14], harmonic structure is generated by using a corpus of 17th-century French dance music. Here, Markov models are used to produce chord progressions. In choosing the corpus, the formal harmonic modeling of different dance forms was examined, among them music by Louis Couperin, Jean Baptiste Lully and Marin Marais. The authors finally decided to use the sarabande and selected 84 examples of this dance form for further examination. The musical material is divided by composer and mode into sub-corpora, while the harmonic progressions are represented

[7] See also the differently structured Markov models described by Chai and Vercoe in [5, p. 3].

as scale degrees. For further processing, the musical material is simplified in different ways, such as by rounding the note lengths or reducing the harmonic structure to triads in root position, where doublings and resolving dissonances are omitted.[8]

Responding to the problems occurring in higher-order Markov models, Ponsford et al. use "smoothed n-grams." A segmentation [14, p. 18ff] of the musical material into phrases and bars conducted for the training of the MM gives better results in the production. The beginning and the end of the particular pieces as well as smaller form sequences are identified in the training corpus so as to allow consideration of context dependencies for the generation. For the production, the transition probabilities are used according to the musical material, depending on the structure: "Numbering phrases effectively makes the data quite a lot sparser, as n-grams at phrase boundaries effectively operate over different sub-corpora, depending on the lengths of the piece they are from." [14, p. 20]. Therefore, for example, it is possible to distinguish between identical phrases (and bars) in chorales of differing length: "However, phrase 6 in a six-phrase piece would not have the same meaning as phrase 6 in an eight-phrase piece." [14, p. 20]. A number of compositions are generated with 3rd and 4th order Markov models, the experiments giving better results in 4th order MM. For the further improvement of the generated material, the authors suggest a correction of the applied Markov models by the user.

3.4 Hidden Markov Models in Algorithmic Composition

Martin Hirzel and Daniela Soukup [9] generated jazz improvisations based on small patterns, which are processed by an HMM. These melodic patterns are entered by the user, form the repertoire of the system and can be transposed to fit provided harmonic progressions. The HMM is trained with "Forest Flower (Sunrise)" by Charles Lloyd and after an input of harmonic progressions serving as the observable sequence, the Viterbi algorithm generates an appropriate succession of melodic patterns – representing the hidden states – that form the compositional output of the system.

Mary Farbood and Bernd Schoner [7] applied hidden Markov models to generate various counterpoints in relation to a given *cantus firmus* (Latin for "fixed melody"). In their work, the authors refer to the sixteenth-century counterpoint and in this context to the so-called first species, one framework of voice leading rules established by the Danish composer and musicologist Knud Jeppesen.[9] On the basis of these voice leading rules, an HMM is created, whose observable sequence is provided by the cantus firmus, and the Viterbi algorithm calculates the hidden states a succession of note values, which form the new counterpoints. An interesting approach in this work is the integration of different rules in one "unifying" transition table: "Each rule is implemented as a probability table where illegal transitions are described by

[8] For an explanation of the different reductions, see [14, p. 14ff].

[9] For the various rules, see Jeppesen's well-known book "Counterpoint: The Polyphonic Vocal Style of the Sixteenth Century" [10].

probability zero. The transition probabilities for generating a counterpoint line are obtained by multiplying the individual values from each table, assuming the rules are independent." [7, p. 3]. Figure 3.11 shows an example of the generation of two independent counterpoints according to a given cantus firmus (bottom).

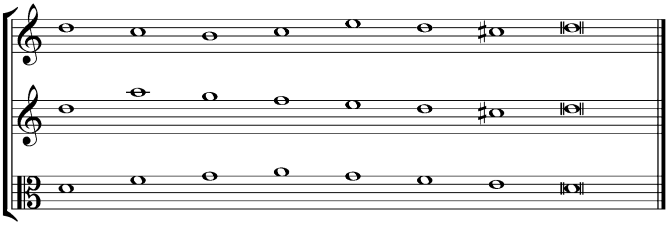


Fig. 3.11 Example generation by Farbood and Schoner according to a given cantus firmus.

3.4.1 A Hierarchical Model

Moray Allan [1] used Markov models and hidden Markov models for the harmonization of given soprano voices. Bach chorales are divided into training and test sets. The musical material is represented by means of a hierarchical segmentation[10] through the tone pitches of all voices, distinguishing amongst others between phrases and bars, the harmonic function also being annotated. The transition probabilities and emission probabilities are determined by frequency analysis of the reference corpus. In comparison to the HMM, Allan also illustrates the possibilities of conventional Markov models up to the 8th order that generate the harmonic function alternatively from the preceding harmonies, the melody tones or a combination of both. In the HMM considered best for the generation, the soprano voice forms the observable sequence in the HMM, while the underlying harmony corresponds to the hidden states. The Viterbi algorithm estimates the underlying state sequence and may also produce a universally acceptable structure in the range of a whole chorale. In contrast, here, conventional Markov models mostly fail to perform this task, because it cannot be estimated how the choice of a particular state transition at a particular time influences the whole structure.[11]

The harmonization is divided into three subtasks[12] that are solved by applying different HMMs. First, a harmonic skeleton is built, in which the concrete pitches and possible doublings are not indicated. The harmonic skeleton represents the hidden states, whereas the notes of the given soprano voice denote the observable sequence. Secondly, on the basis of the harmonic symbols as the observable sequence,

[10] For the representation of the musical material, cf. the scheme in [1, p. 18].

[11] See also Nearest Neighbor Heuristic in chapter 10.

[12] Cf. "Final harmonization model" in [1, p. 43ff].

another HMM generates concrete chords that correspond to the hidden states. Finally, a third HMM model is responsible for the ornamentation, whereas compound symbols, indicating the harmonic symbol and the notes of the current and next beat, denote the observable sequence. Each quaver associated to a harmonic symbol is represented as four distinct 1/16 notes, which form the hidden states and can – if partly altered by the HMM – result in various ornamentations. Figure 3.12 shows different examples of this three-stage generation process.

Fig. 3.12 Part of a three-stage harmonization process by Moray Allan [1, p. 49ff] according to the first chorale lines of "Dank sei Gott in der Höhe," BWV 288. With kind permission of Moray Allan.

3.4.2 Stylistic Classification

Wei Chai and Barry Vercoe [5] applied hidden Markov models as a means to distinguish the stylistic differences in music styles of different provenience. Although this approach is not used for the generation of musical structure, it contains interesting aspects of representation, worth to be mentioned in the context of this chapter. Chai and Vercoe represent the musical material in four different ways: First, absolute pitch representation within one octave space, starting with c1; secondly, absolute pitch representation with duration, where information on various durations is provided by eventually repeating each note multiple times; thirdly, interval representation, in which the pitches are converted into a sequence of intervals; fourthly, contour representation – a symbolic representation of the melodic contour, where the interval changes are denoted with the following symbols: 0 for no change, $+/-$ for ascending/descending 1 or 2 semitones, $++/--$ for ascending/ descending 3 or more semitones. The authors work with 16 different HMM, which are constructed out of different numbers of hidden states (2 to 6) in combination with four basic structures (see figure 3.13): "(a) A strict left-right model, each state can transfer to itself and the next one state. (b) A left-right model, each state can transfer to itself and any state right to it. (c) Additional to (b), the last state can transfer to the first state. (d) A fully connected model." [5, p. 3].

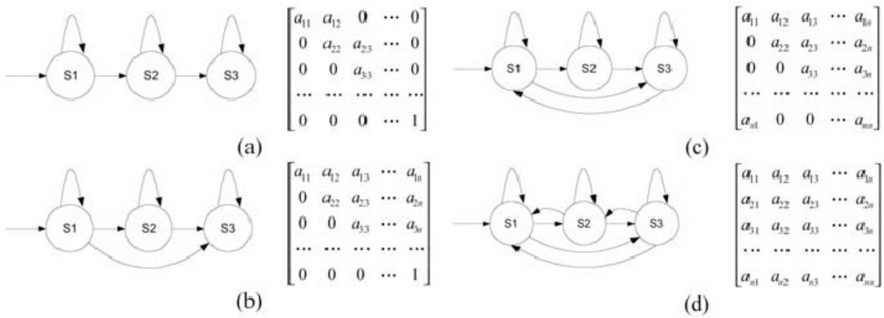


Fig. 3.13 Different types of Markov models by Chai et al. [5, p. 3]. With kind permission of Wei Chai.

For the analysis, the authors decide to use a corpus of monophonic folk music melodies from Ireland, Germany and Austria. The different folk melodies are separated into a training set (70%) and a test set (30%). The parameters of the HMMs are trained in regard to each country with the Baum–Welch algorithm, whereas the Viterbi algorithm is used to assign a melody from the test set to a particular HMM, thus allowing for a stylistic classification. The results point out structural similarities between German and Austrian folk songs and further demonstrate that stylistic classification may best be performed by HMMs with structure (a) or (b), whereas the number of states is of less importance.

3.5 Synopsis

Markov models are, like generative grammars, *substitution systems*, and due to their structure only allow the description of context dependencies through the transition probabilities of symbols in direct succession. These formalisms were originally developed in the context of natural language processing and are best suited to model one-dimensional symbol sequences. This structural feature of MMs does not correspond to musical information, which mostly adds a vertical dimension through layers, i.e. interconnected voices. Of course, a model defining all possible vertical constellations as single states could overcome this restriction, but if the required number of states is considered, this approach turns out to be a rather theoretical solution. A possible way to frame dependencies within a vertical structure consists in the application of hidden Markov models as coupled stochastic processes. Within the HMM, the Viterbi algorithm provides a suitable tool for the generation of an overall favorable structure, a demand, which can hardly be accomplished by the conventional formalism, except by approaches using several Markov models in a hierarchical interconnected structure. Since especially higher orders show a large size of the transition tables, these are mostly generated by analyzing an underlying corpus, leading to the fact that Markov models are in the majority of the cases used in the field of style imitation, with some restrictions: The danger of an obvious randomness with lower-order Markov models and also the frequent occurrence of an over-generation by models of higher order, meaning that large sections of the corpus are simply "re-generated." If the consistency of the corpus restricts the establishment of a higher-order Markov model, smoothed n-grams can be applied that use lower-order transitions for the generation of higher-order transition probabilities. An often overlooked deficiency in higher-order Markov models lies in their inability to indicate information which is provided in lower-order models. So, for example [15, p. 196–197] the symbol string AFGBBFGCFGDFG#EFG can be exhaustively described and therefore regenerated with a 3rd order Markov model, because there are no equal groups composed of three successive symbols in the corpus. This 3rd order Markov model provides an accurate description, which can actually be used for a regeneration of the corpus, but the apparent fact that an F is always followed by a G cannot – in contrast to a 1st order Markov model – be depicted.

Despite these disadvantages, Markov models are well suited to certain musical tasks. The type and quality of the output will depend largely on the properties of the corpus and can also be predicted very well in comparison to other procedures such as neural networks or cellular automata. Besides for style imitation, composers like Xenakis use Markov models in an innovative way, demonstrating their applicability also for the field of computer assisted composition, where this formalism is in general employed only rarely.

References

1. Allan M (2002) Harmonising chorales in the style of Johann Sebastian Bach. Thesis, University of Edinburgh, 2002
2. Basharin GP, Langville AN, Naumov VA (2004) The life and work of A. A. Markov. http://www.maths.tcd.ie/ nhb/talks/Markov.pdf Cited 24 Jul 2005
3. Bronstein IN, Semendjajew KA, Musiol G, Mühlig H (2000) Taschenbuch der Mathematik, 5th edn. Verlag Harri Deutsch, Thun und Frankfurt am Main. ISBN 3-8171-2015-X
4. Brooks FP, Hopkins AL, Neumann PG, Wright WV (1993) An experiment in musical composition. In: Schwanauer SM, Levitt DA (eds) Machine models of music. MIT Press, Cambridge, Mass. ISBN 0-262-19319-1
5. Chai W, Vercoe B (2001) Folk music classification using hidden Markov models. In: Proceedings International Conference on Artificial Intelligence, 2001
6. Dodge C, Jerse TA (1997) Computer music: synthesis, composition, and performance, 2nd edn. Schirmer Books, New York. ISBN 0-02-864682-7
7. Farbood M, Schoner B (2001) Analysis and synthesis of Palestrina-style counterpoint using Markov chains. In: Proceedings of International Computer Music Conference. International Computer Music Association, San Francisco
8. Hiller L, Isaacson L (1993) Musical composition with a high-speed digital computer. In: Schwanauer SM, Levitt DA (eds) Machine models of music. MIT Press, Cambridge, Mass. ISBN 0-262-19319-1
9. Hirzel M, Soukup D (2000) Project Writeup for CSCI 5832 natural language processing. University of Colorado, 2000
10. Jeppesen K (1992) Counterpoint: the polyphonic vocal style of the sixteenth century. Dover Publications, New York. ISBN 04862736X
11. Jones K (1981) Compositional applications of stochastic processes. Computer Music Journal 5/2, 1981
12. Knab B (2000) Erweiterungen von Hidden-Markov-Modellen zur Analyse ökonomischer Zeitreihen. Dissertation, Mathematisch-Naturwissenschaftliche Universität Köln
13. Olson H (1967) Music, physics and engineering, 2nd edn. Dover Publications, New York. ISBN 0-486-21769-8
14. Ponsford D, Wiggins G, Mellish C (1999) Statistical learning of harmonic movement. Journal of New Music Research, 1999
15. Todd PM, Loy DG (eds) (1991) Music and connectionism. MIT Press, Cambridge, Mass. ISBN 0-262-20081-3
16. Xenakis I (1992) Formalized music. Thought and mathematics in music. Pendragon, Stuyvesant, NY. ISBN 0-945193-24-6