hyväksymispäivä arvosana

arvostelija

Markovin mallit algoritmisessa säveltämisessä

Aki Utoslahti

Helsinki 15.5.2019

Kandidaatintutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen osasto

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen tiedekunta	Tietojenkäsittelytieteen osasto

Tekijä — Författare — Author
Aki Utoslahti

Työn nimi — Arbetets titel — Title

Markovin mallit algoritmisessa säveltämisessä

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma	15.5.2019	22 sivua + 1 liitesivua

Tiivistelmä — Referat — Abstract

Tutkielman tarkoituksena on toimia katsauksena Markovin mallien hyödyntämiseen algoritmisessa säveltämisessä. Lähdeaineistona toimivat alan uraauurtavat tutkimukset ja perusoppikirjat. Aluksi perehdytään Markovin mallien perusperiaatteisiin ja -ominaisuuksiin, joita täydentämään esitellään yksinkertainen menetelmä tuottaa musiikkia Markovin ketjujen avulla. Koska osoittautuu, että Markovin mallin perinteisillä soveltamismenetelmillä ei voida tuottaa mielenkiintoista uutta musiikkia, kohdistetaan erityinen mielenkiinto niiden hyödyntämiseen yhdessä muiden algoritmisen säveltämisen menetelmien kanssa. Tutkielmaan onkin siksi poimittu joitakin merkittäviä esimerkkejä siitä, miten Markovin malleilla on voitu edesauttaa laadullisesti hyvien tulosten saavuttamista niin tietokoneavusteisessa kuin autonomisessa säveltämisessä.

ACM Computing Classification System (CCS):
Applied computing → Arts and humanities → Sound and music computing
Mathematics of computing → Probability and statistics → Stochastic processes → Markov processes

Avainsanat — Nyckelord — Keywords
algoritminen säveltäminen, Markovin ketju, Markovin piilomalli

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Sisältö

1 Johdanto

Musiikin on perinteisesti ajateltu edellyttävän ihmislähtöistä taiteellisuutta, oli sitten kyse sen tulkinnasta tai säveltämisestä. Historian kuluessa sävellystyötä helpottamaan on kehitetty lukuisia formalisoituja keinoja [1] ja siksi voidaankin todeta, että musiikki ei ole peräisin ainoastaan taiteellisesta luovuudesta. Tämän vuoksi onkin tarpeen tutkia miten ja missä määrin sävellystyötä pystytään formalisoimaan ja voidaanko uudenlaista musiikkia säveltää pelkästään tietokoneelle formalisoiduin keinoin.

Satunnaisuuteen pohjaavien sävellyskeinojen kehittäminen on kiinnostanut säveltäjiä erityisesti tietokoneiden keksimisen jälkeen [2]. Pelkkään satunnaisuuteen perustuvat tulokset ovat kuitenkin erittäin harvoin kovinkaan kiinnostavia taiteellisesta näkökulmasta, minkä vuoksi satunnaisuuteen perustuville sävellysprosesseille on pystyttävä asettamaan rajoitteita tavalla tai toisella. Satunnaisuuteen ja todennäköisyyksiin pohjautuvat stokastiset prosessit ovatkin eräs oivallinen keino näiden rajoitteiden mahdollistamiselle.

Tietokoneilla tapahtuvan algoritmisen säveltämisen voidaan katsoa syntyneen vuonna 1956, Hillerin ym. toimesta, kun ensimmäinen kokonaan tietokoneella sävelletty musiikillinen teos, *Illiac Suite* -jousikvartetto, sai maailmanensi-iltansa [1]. Erityisen kiinnostavan jousikvartetossa tekee sen neljäs osa, joka on sävelletty kokonaan stokastisia prosesseja, nimenomaisesti Markovin malleja, hyödyntäen.

Tutkielman tarkoituksena on perehtyä siihen, miten Markovin mallien avulla voidaan tuottaa musiikkia ja ovatko niiden avulla saavutetut satunnaisuuden rajoitteet riittäviä kiinnostavien sävellysten tuottamiseksi. Pyrkimyksenä on myös valottaa Markovin mallien avulla saavutettua lisäarvoa, kun niitä käytetään yhdessä muiden algoritmisen säveltämisen menetelmien kanssa. Tutkielman luku kaksi käsittelee Markovin ketjujen ja piilomallien teoriataustaa. Luku kolme puolestaan esittelee erään yksinkertaisen menetelmän algoritmiseen säveltämiseen Markovin ketjujen avulla. Neljännessä luvussa tarkastellaan erilaisia tuloksia, joita on saatu Markovin mallien soveltamisella sekä yhdistelyllä muihin algoritmisen säveltämisen menetelmiin. Viimeisessä luvussa esitetään johtopäätökset Markovin mallien hyödyllisyydestä algoritmisessa säveltämisessä.

2 Markovin mallit

Markovin malleilla on merkittävä osa stokastisten prosessien tutkimuksessa. Ne ovat avain satunnaisten prosessien mallintamiseen aina uhkapelurin alati muuttuvasta onnesta, satunnaisen pariutumisen ja mutaatioiden aiheuttamaan geneettiseen vaihteluun saakka [3]. Tässä luvussa perehdytään algoritmiseen säveltämiseen soveltuvien mallien, Markovin ketjujen ja piilomallien, teoriataustaan.

2.1 Markovin ketju

Markovin ketjun kehitti vuonna 1906 venäläinen matemaatikko Andrei Markov tutkiessaan kirjainten esiintymistodennäköisyyksiä Aleksandr Puškinin kirjassa Jevgeni Onegin [1]. Kyseessä on ehdolliseen todennäköisyyteen perustuva matemaattinen järjestelmä [2], jolla voidaan mallintaa tapahtumien sarjan käytöstä [3].

Määritelmä 1 *Stokastinen prosessi on joukko ajasta riippuvia satunnaismuuttujia* $(X_t)_{t \in T}$*. Satunnaismuuttujat saavat arvoja tila-avaruudessa* S *ja* t *on parametrijoukkoon tai aikaskaalaan* T *kuuluva aikaindeksi [4].*

Määritelmä 2 *Stokastinen ketju on stokastinen prosessi, joka on diskreettiaikainen ja saa arvoja diskreetissä tila-avaruudessa [1].*

Määritelmä 3 *Olkoon* $(X_n)_{n \in \mathbb{N}}$ *stokastinen ketju. Prosessin* $(X_n)_{n \in \mathbb{N}}$ *sanotaan olevan markovilainen tai noudattavan Markovin ominaisuutta, kun kaikille* $n \geq 1$ *satunnaismuuttujan* X_{n+1} *todennäköisyysjakauman määrittää prosessin tila* X_n *ajanhetkellä* n*, eikä se riipu aikaisemmista arvoista* $X_k, k = 0, 1, \dots, n-1$ *[4].*

Markovin ketju on siis stokastinen malli, joka koostuu äärellisestä tilojen joukosta ja näiden tilojen välisistä tilasiirtymistä. Tilat voivat käytännössä kuvata mitä tahansa tapahtumia tai symboleita, kun taas tilasiirtymät ovat muutoksia niiden välillä [3]. Markovin ominaisuus käytännössä tarkoittaa sitä, että ketjun käytöksen tulee olla muistiton (engl. memory-less), eli seuraava valittava tila voi riippua ainoastaan nykyisestä tilasta. [4]

Markovin ketjussa seuraava tila X_{t+1} riippuu nykyisestä tilasta X_t. Ajoille t_m ja t_{m+1} pätee todennäköisyys [1]

$$Pr(X_{t_{m+1}} = j | X_{t_m} = i) = p_{ij}(t_m, t_{m+1}). \tag{1}$$

Lauseke ilmaisee siis siirtymätodennäköisyyden sille, että tietyllä ajanhetkellä t_m päädytään lähtötilasta X_{t_m} kohdetilaan $X_{t_{m+1}}$.

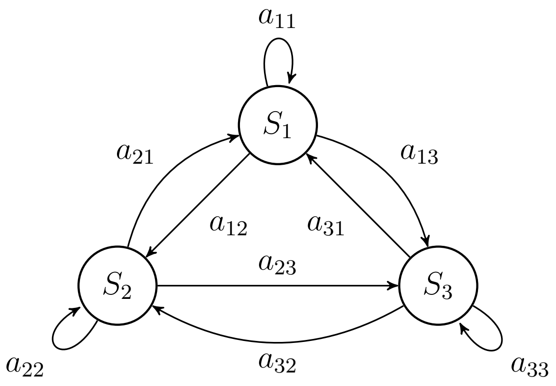
Markovin ketjua kuvataan joko tilasiirtymäverkkona (engl. transitional graph) tai tilasiirtymämatriisina (engl. transitional matrix) [1]. Tilasiirtymäverkossa (kuva 2.1a) solmut vastaavat Markovin ketjun tiloja ja kaarien painot siirtymätodennäköisyyksiä. Vastaavat siirtymätodennäköisyydet löytyvät myös tilasiirtymämatriisista (kuva 2.1b). Tilasiirtymämatriisi P on $S \times S$ matriisi, jolle pätee [4]

$$P = [p_{ij}]_{i,j \in S} = [Pr(S_1 = j |\ S_0 = i)]_{i,j \in S}. \tag{2}$$

Matriisin solut ilmaisevat siis todennäköisyyden tilasiirtymälle alkutilasta i kohdetilaan j. Erityisesti siis lähtötilaa i vastaa matriisin rivi-indeksi ja kohdetilaa j matriisin sarakeindeksi. Lisäksi jokainen tilasiirtymämatriisin rivi on todennäköisyysvektori, jolle pätee [4]

$$\forall i \in S : \sum_{j \in S} P_{ij} = 1, \tag{3}$$

eli jokaisen tilasiirtymämatriisin rivin todennäköisyyksien summa on 1.



(a)

$$P = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \end{array}$$

(b)

Kuva 2.1: Kolmetilaisen Markovin ketjun esitysmuodot siirtymäverkkona (a) ja siirtymämatriisina (b). Parametrit: S – ketjun tilat, a – ketjun tilojen väliset siirtymätodennäköisyydet.

Määritelmä 4 *Markovin ketju on kolmikko (S, Q, A), jossa:*

1. *S on äärellinen tilojen joukko*
2. *Q on tilojen $s \in S$ alkeistodennäköisyyksien jakauma*
3. *$A : S \times S \to Pr(S)$ on tilasiirtymäfunktio, joka antaa todennäköisyyden sille, että tilasta s siirrytään tilaan s'.*

Markovin ketju voidaan muodostaa joko ohjelmoituna tai opittuna [5]. Molemmissa tapauksissa muodostaminen alkaa tilajoukon määrittämisellä. Tekstiaineiston tapauksessa tilajoukoksi voidaan määritellä esimerkiksi käytetyt yksittäiset merkit, sanat tai lauseet. Kun tilajoukko on määritelty, voidaan siirtymätodennäköisyydet joko opettaa opetusaineiston avulla tai ohjelmoida itse. Opetusaineiston tapauksessa yksinkertaisinta on kerätä matriisiin opetusaineistossa esiintyvät tilasiirtymät, eli kuinka monta kertaa esimerkiksi merkkiä A seurasi merkki B ja niin edelleen. Kun tilasiirtymät on laskettu, voidaan esiintymämatriisi normalisoida tilasiirtymämatriisiksi [4].

Kuvassa 2.2 on käytetty opetusaineistona satunnaisesti generoitua merkkijonoa, joka koostuu merkeistä A, B ja C, eli mallin tilajoukko on $S = \{A, B, C\}$. Kuvaan 2.2a on laskettu jokaisen merkin ja merkkisiirtymän esiintymät opetusaineistossa. Kuvaan 2.2b puolestaan on normalisoitu merkkien ja merkkisiirtymien esiintymät todennäköisyyksiksi. Esimerkiksi merkistä C siirryttiin kerran merkkiin A ja 4 kertaa merkkiin B, jolloin siirtymiä vastaavat todennäköisyydet $C \to A = \frac{1}{5}$ ja $C \to B = \frac{4}{5}$. Kun Markovin ketju on muodostettu, voidaan sen avulla tuottaa uusia tilajonoja, jotka noudattavat mallia. Ensiksi on kuitenkin valittava alkutila, tila, josta tuottaminen alkaa. Alkutila voidaan valita monilla erilaisilla menetelmillä, yleisimmän ollessa todennäköisyysperustainen valinta kaikkien tilojen tilajakauman mukaisesti

[5]. Kun alkutila on valittu, voidaan uusia tilajonoja tuottaa satunnaislukujen avulla tilasiirtymämatriisia noudattaen, kunnes haluttu pituus tuotettavalle tilajonolle on saavutettu.

$$\begin{array}{ccc} A & B & C \\ \end{array}$$
$$\begin{bmatrix} 9 & 6 & 5 \end{bmatrix} \qquad \begin{bmatrix} 9/20 & 3/10 & 1/4 \end{bmatrix}$$

$$\begin{array}{c|c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 4 & 2 & 3 \\ 3 & 0 & 2 \\ 1 & 4 & 0 \end{bmatrix} \end{array} \qquad \begin{array}{c|c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 4/9 & 2/9 & 1/3 \\ 3/5 & 0 & 2/5 \\ 1/5 & 4/5 & 0 \end{bmatrix} \end{array}$$

(a) (b)

Kuva 2.2: Merkkijonon AACBCBCBAABAACBAACAB tilojen ja tilasiirtymien esiintymät (a) sekä niistä normalisoitu alkeistodennäköisyyksien jakauma ja tilasiirtymämatriisi (b). (Huom. tilasiirtymämatriiseissa huomioidaan siirtymät merkkijonon ensimmäisestä merkistä alkaen.)

Markovin ketjun muistia on mahdollista laajentaa niin, että se ulottuu nykyisestä solmusta $n-1$ solmua taaksepäin, jolloin on tapana puhua astetta n olevasta Markovin ketjusta (engl. n^{th} order Markov chain) [1]. Esimerkiksi toisen asteen Markovin ketjussa siirtymätodennäköisyyteen vaikuttaa nykyistä solmua edeltävä solmu, kolmannen asteen ketjussa nykyistä solmua edeltävät kaksi solmua ja niin edelleen. Korkeamman asteen Markovin ketjujen esittäminen tilasiirtymäverkkona tai -matriisina on kuitenkin haastavaa, sillä esimerkiksi matriisissa tarvitaan tällöin $n+1$ ulottuvuutta ja verkon käyttäminen on mahdotonta. Jokaiselle asteen n Markovin ketjulle on kuitenkin olemassa sitä vastaava ensimmäisen asteen Markovin ketju, toisin sanoen Markovin ominaisuus säilyy vaikka Markovin ketjun astelukua kasvatetaan. Yksinkertaisinta tapaa esittää nämä vastaavat ensimmäisen asteen Markovin ketjut kutsutaan n-grammiksi (engl. n-gram). N-grammissa tilasiirtymämatriisin vasen sarake koostuu n tilan mittaisista sekvensseistä, joista ilmenevät kaikki tilasiirtymien todennäköisyyksiin vaikuttavat dimensiot. Kuvassa 2.3 on esitetty opetusaineistona käytetyn satunnaisesti generoidun merkkijonon 2- ja 3-asteiset Markovin ketjut.

Kun Markovin ketjun astelukua kasvatetaan, ketjun avulla tuotettu aineisto alkaa muistuttaa yhä enemmän opetusaineistoa, jopa toistaen pitkiä identtisiä sekvenssejä sellaisenaan. Vastaavasti ensimmäisen asteen Markovin ketjun tuotokset voivat niiden korkean satunnaisuuden asteen vuoksi olla epämusikaalisia ja niiden melodia voi vaellella päämäärättömästi [5]. Ketjun asteluvun valinnassa tulee siis tehdä kompromissi mallin stabiiliuden ja sopivuuden välillä. Markovin ketjun asteluvun kasvattamisesta voi seurata myös ongelmia, sillä mikäli tietty n mittainen sekvenssi

$$\begin{array}{c|ccc} & A & B & C \\ \hline AA & 0 & 1/4 & 3/4 \\ AB & 1 & 0 & 0 \\ AC & 1/3 & 2/3 & 0 \\ BA & 1 & 0 & 0 \\ BC & 0 & 1 & 0 \\ CA & 0 & 1 & 0 \\ CB & 1/2 & 0 & 1/2 \end{array}$$

(a)

$$\begin{array}{c|ccc} & A & B & C \\ \hline AAB & 1 & 0 & 0 \\ AAC & 1/3 & 2/3 & 0 \\ ABA & 1 & 0 & 0 \\ ACA & 0 & 1 & 0 \\ ACB & 1/2 & 0 & 1/2 \\ BAA & 0 & 1/3 & 2/3 \\ BCB & 1/2 & 0 & 1/2 \\ CBA & 1 & 0 & 0 \\ CBC & 0 & 1 & 0 \end{array}$$

(b)

Kuva 2.3: Merkkijonon AACBCBCBAABAACBAACAB toisen (a) ja kolmannen asteen (b) Markovin ketjut n-grammina esitettynä.

ei esiinny opetusaineistossa, ei se esiinny myöskään ketjun ulostulossa. Eräs keino ongelman välttämiseen ovat tasoitetut n-grammit (engl. smoothed n-grams), jotka hyödyntävät matalamman asteen siirtymätodennäköisyyksiä korkeampien asteiden siirtymätodennäköisyyksien määrittelyyn. Tällöin puuttuvat asteen n siirtymätodennäköisyydet voidaan interpoloida alemmilla asteilla $n-1, n-2, \ldots$ [1].

2.2 Markovin piilomalli

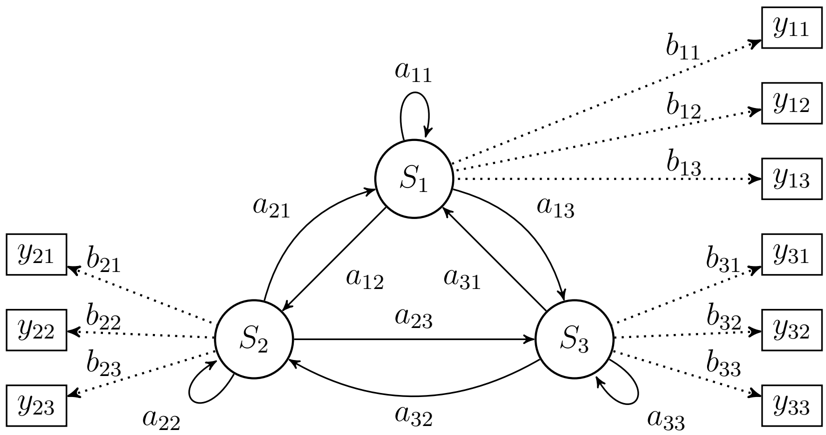
Markovin ketjun tuottaman symbolijonon perusteella voidaan yksiselitteisesti päätellä millaisia tilasiirtymiä tuottamisen aikana on tapahtunut [1]. Markovin piilomalli on Markovin ketju, jossa ainoastaan prosessin ulostuloa voidaan tarkkailla, kun taas mallin sisäinen tila ja tilasiirtymät ovat piilotettuja. Piilomallin tapauksessa ulostulo tuotetaan todennäköisyysperustaisten emissiofunktioiden avulla. Piilomalli on siis paritettu stokastinen prosessi, sillä tilojen välille on määritetty siirtymätodennäköisyydet ja jokaisella tilalla on omat emissiotodennäköisyytensä. Kuvassa 2.4 on esitetty kolmetilainen Markovin piilomalli, jossa jokaisella tilalla on kolme mahdollista emissiota. Tapauksesta riippuen emissiot voivat olla täysin erilliset, mutta on myös mahdollista, että eri tilojen eri emissiot tuottavat samoja ulostulosymboleita.

Piilomalli mahdollistaa erilaisten oletuksien ja päätelmien tekemisen mallin sisäisestä tilasta, joista suurin osa voidaan ratkaista seuraavilla algoritmeilla: [1]

- Suodatus (engl. filtering, forward algorithm) laskee ulostulona esiintyvän sekvenssin todennäköisyyden, kun mallin alkeistodennäköisyyksien jakauma sekä tilasiirtymien ja niihin liittyvien emissioiden todennäköisyydet ovat tiedossa.
- *Viterbi*-algoritmi selvittää todennäköisimmän piilotettujen tilojen sekvenssin,

Viterbi-polun, jolla ulostulona saatu sekvenssi on voitu tuottaa.

- *Baum-Welch* -algoritmia käytetään löytämään todennäköisimmät piilomallin sisäiset parametrit ulostulona saadun sekvenssin perusteella.



Kuva 2.4: Markovin piilomalli. Mallissa on kolme tilaa, joilla jokaisella on kolme mahdollista emissiota. Parametrit: S – ketjun tilat, y – tilojen emissiot, a – siirtymätodennäköisyydet, b – emissiotodennäköisyydet

3 Säveltäminen Markovin ketjuilla

Markovin mallien soveltaminen algoritmisessa sävellystyössä on jatkunut aivan viime päiviin asti 1950 -luvun puolivälistä lähtien. Katsaus historiaan [1] osoittaa Hillerin ym. ja Xenakiksen toimineen näiden stokastisten sävellysprosessien uranuurtajina, lukuisien muiden tutkijoiden jatkaessa heidän jalanjäljissään. Tässä luvussa esitellään yksinkertainen menetelmä opetusaineistoon perustuvien melodioiden tuottamisesta Markovin ketjujen avulla.

Algoritmisessa säveltämisessä Markovin ketjut tuottavat musiikillisia symboleita [5]. Nämä symbolit voivat olla esimerkiksi nuotteja, rytmejä, sointuja tai edellisten sekvenssejä, jotka siis Markovin ketjun tapauksessa vastaavat ketjun tiloja. Yleinen menetelmä on määrittää Markovin ketjun tilat vastaamaan yksittäisiä nuotteja. Opetusaineiston tapauksessa tämä vaatii opetusaineiston analysoinnin ja nuottien sävelkorkeuden jakauman ja tilasiirtymämatriisin muodostamista aliluvussa 2.1 kuvatulla tavalla. Kuvassa 3.1 on analysoinnin tuloksena muodostettu malli J.S. Bachin sävellyksessä Air (liite A) esiintyvän viuluosion melodian sävelkorkeuksista. Mallin pohjalta voidaan tuottaa uusi melodia seuraavilla askelilla:

1. Generoi satunnaisluku ja valitse sen perusteella alkutila mallin alkeistodennäköisyyksien jakaumasta (kuva 3.1a)

$$
\begin{array}{cccccccccccccc}
a_4 & a_4\sharp & h_4 & c_5 & c_5\sharp & d_5 & d_5\sharp & e_5\sharp & f_5 & g_5 & g_5\sharp & a_5 & h_5 & c_6 \\
\end{array}
$$
$$
\begin{bmatrix} 9/154 & 1/154 & 15/154 & 3/154 & 3/22 & 25/154 & 1/154 & 10/77 & 1/7 & 19/154 & 1/77 & 5/77 & 5/154 & 1/154 \end{bmatrix}
$$

(a)

$$
\begin{array}{c|cccccccccccccc|}
 & a_4 & a_4\sharp & h_4 & c_5 & c_5\sharp & d_5 & d_5\sharp & e_5\sharp & f_5 & g_5 & g_5\sharp & a_5 & h_5 & c_6 \\
\hline
a_4 & 1/9 & & 1/9 & 1/9 & 1/18 & 2/9 & & 1/9 & 1/18 & & & 2/9 & & \\
a_4\sharp & & & 1 & & & & & & & & & & & \\
h_4 & 4/15 & 1/15 & 1/15 & & 2/5 & 1/15 & & 1/15 & & & & & 1/15 & \\
c_5 & & & 2/3 & & & & & 1/3 & & & & & & \\
c_5\sharp & 2/21 & & 1/3 & & 2/21 & 1/3 & & 1/21 & 1/21 & 1/21 & & & & \\
d_5 & & & 2/49 & & 23/49 & 10/49 & & 6/49 & 6/49 & & & & 2/49 & \\
d_5\sharp & & & & & & & & & & & & 1 & & \\
e_5\sharp & & & 1/20 & & & 7/20 & 1/20 & 3/20 & 1/5 & 3/20 & & 1/20 & & \\
f_5 & & & & 1/22 & 1/22 & & & 3/11 & 3/11 & 2/11 & 1/22 & 1/11 & 1/22 & \\
g_5 & 1/19 & & 1/19 & & & & & 1/19 & 6/19 & 7/19 & & 1/19 & & \\
g_5\sharp & & & & & & & & 1/2 & & & & 1/2 & & \\
a_5 & 1/10 & & & 1/10 & & 1/5 & & & 1/10 & 3/10 & 1/10 & & & 1/10 \\
h_5 & & & & & & 1/5 & & & & 1/5 & & 2/5 & 1/5 & \\
c_6 & & & & & & & & & & & & & 1 & \\
\end{array}
$$

(b)

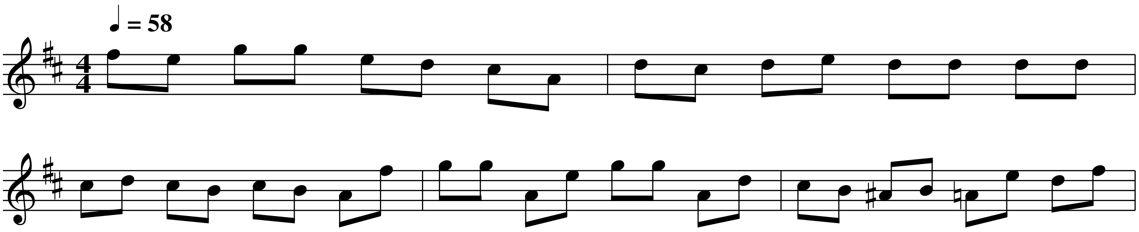
Kuva 3.1: J.S. Bach: Air -teoksen sävelkorkeuksien alkeistodennäköisyyksien jakauma (a) ja tilasiirtymämatriisi (b), jossa tyhjät solut ilmaisevat siirtymätodennäköisyyttä 0.

2. Generoi satunnaisluku ja valitse sen perusteella seuraava tila nykyisen tilan todennäköisyysvektorista, tilasiirtymämatriisista (kuva 3.1b). Toista tätä askelta, kunnes haluttu pituus tuotettavalle melodialle on saavutettu.

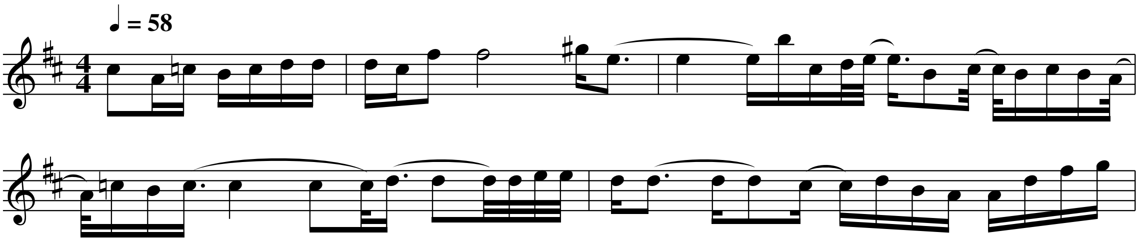
Kuvassa 3.2a on esimerkki edellä kuvatulla tavalla tuotetusta melodiasta. Melodiasta voidaan kuitenkin havaita, että siinä ei ole lainkaan rytmillisiä vaihteluita. Kuvassa 3.3 on saman teoksen rytmillisistä vaihteluista muodostettu malli, jonka avulla voimme laajentaa edellistä sävellysmenetelmää tuottamaan myös rytmillisiä vaihteluita:

1. Generoi satunnaisluku ja valitse sen perusteella alkutila sekä melodisen (kuva 3.1a) että rytmisen mallin (kuva 3.3a) alkeistodennäköisyyksien jakaumista.

2. Generoi satunnaisluku ja valitse sen perusteella seuraava tila sekä melodisen (kuva 3.1b) että rytmisen mallin (kuva 3.3b) tilasiirtymämatriiseista. Toista tätä askelta, kunnes haluttu pituus tuotettavalle melodialle on saavutettu.



(a)



(b)

Kuva 3.2: J.S. Bach - Air teoksen pohjalta generoidut esimerkit: melodia ilman rytmillisiä muutoksia (a) ja melodian ja rytmin yhdistelmä (b).

Kuvassa 3.2b on esimerkki laajennetulla, rytmin huomioivalla, menetelmällä tuotetusta melodiasta. Esimerkin melodia alkaa jo muistuttaa musiikkia rytmin tuoman vaihtelun ansiosta, mutta sen loppua voitaisiin vielä parantaa. Opetusaineistona käytetty J.S. Bachin teos Air noudattaa D -duurisävelasteikkoa ja sen osoituksena päättyy nuottiin d_5. Yleisesti klassisen musiikin teokset päättyvät joko noudattamansa sävelasteikon ensimmäiseen tai viidenteen säveleen. Toimiva keino parantaa tuotettavan sävellyksen harmoniaa olisi siis asettaa se loppumaan ensimmäiseen vastaantulevaan ykkös- tai viitossäveleen, kun tietty prosenttiosuus tavoitellusta sävellyksen pituudesta on saavutettu [6].

Viimeinen huomioitava piirre ovat musiikissa ilmenevät tauot. Esimerkkien opetusaineistona käytetyssä teoksessa ei ollut yhtään taukoa, joten on ilmeistä ettei niitä päätynyt myöskään esimerkkeihin. Mikäli opetusaineisto kuitenkin sisältää taukoja, tulisi ne huomioida jo mallin rakentamisvaiheessa [3]. Eräs menetelmä taukojen huomioimiseen on laskea taukojen ja nuottien välinen todennäköisyysjakauma, jolloin jokaisella tuotosaskeleella voidaan generoida satunnaisluku, joka määrittää onko seuraava tuotettava symboli nuotti vai tauko. Lisäksi erilaisten taukojen ilmenemiskertojen perusteella voitaisiin määrittää taukoja koskeva alkeistodennäköisyyksien jakauma, jolloin seuraavaksi generoidulla satunnaisluvulla määritettäisiin seuraavaksi tuotettavan tauon pituus. Edellä kuvatussa tapauksessa algoritmiseen säveltämiseen käyttämämme ketju olisi siis hierarkinen Markovin ketjuista koostuva Markovin ketju (kuva 3.4).

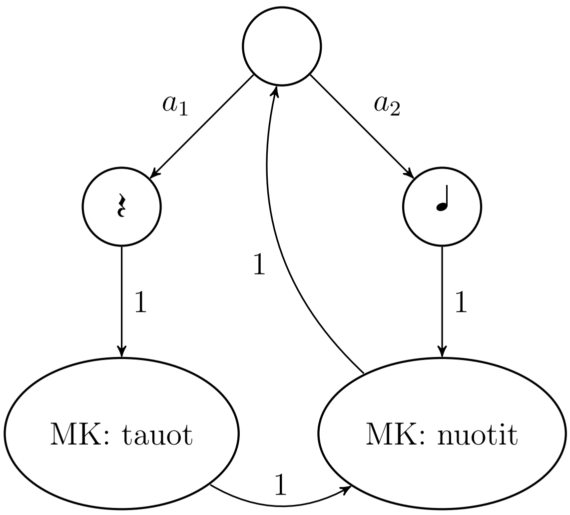
𝅝	𝅗𝅥	𝅘𝅥.	𝅘𝅥	𝅘𝅥𝅮.	𝅘𝅥𝅮	𝅘𝅥𝅯	𝅘𝅥𝅰
1/155	6/155	7/155	9/155	2/155	21/155	87/155	22/155

(a)

	𝅝	𝅗𝅥	𝅘𝅥.	𝅘𝅥	𝅘𝅥𝅮.	𝅘𝅥𝅮	𝅘𝅥𝅯	𝅘𝅥𝅰
𝅝						1		
𝅗𝅥	1/11			6/11			4/11	
𝅘𝅥.						2/7	5/7	
𝅘𝅥		1/9	1/9	1/9		1/9	4/8	1/9
𝅘𝅥𝅮.							1/2	1/2
𝅘𝅥𝅮		1/21			1/21	2/7	4/7	1/21
𝅘𝅥𝅯		4/87	2/29	4/87		3/29	2/3	2/29
𝅘𝅥𝅰				1/22	1/22	1/11	5/22	13/22

(b)

Kuva 3.3: J.S. Bach: Air -teoksen sävelkestojen alkeistodennäköisyyksien jakauma (a) ja tilasiirtymämatriisi (b), jossa tyhjät solut ilmaisevat siirtymätodennäköisyyttä 0.



Kuva 3.4: Musiikissa ilmenevät tauot huomioivan Markovin ketjun (MK) periaate. Jokainen tuotettu tauko aiheuttaa siirtymän nuotteja tuottavaan Markovin ketjuun ja jokainen tuotettu nuotti aiheuttaa siirtymän takaisin alkutilaan. Nuotteja tuottavan ketjun edeltävänä tilana toimii edellinen tuotettu nuotti. Parametrit: a - taukojen ja nuottien todennäköisyysjakauma.

4 Markovin mallien sovellukset

Markovin malleilla on algoritmisen säveltämisen kontekstissa joitakin mielenkiintoisia sovelluksia. Myös niiden yhdistely muihin algoritmisen säveltämisen menetelmiin nk. hybridimenetelmissä on yleistynyt viime vuosikymmeninä [5]. Tässä luvussa käsitellään joitakin merkittäviä Markovin mallien sovelluksia ja hybridimenetelmiä sekä niiden avulla saavutettuja tuloksia.

4.1 Rajoitteet

Algoritmisessa säveltämisessä on useita sovelluskohteita, jotka edellyttävät rajoitteiden sisällyttämistä sävellysprosessiin [7]. Rajoitteista hyödytään erityisesti musiikillisen materiaalin tai tyylin imitoinnissa. Rajoitteet ovat kuitenkin ongelmallisia, sillä kontrolliperustaisina ne yleisessä tapauksessa rikkovat Markovin ominaisuutta äärellisen muistin osalta. Ongelmaa on yritetty ratkaista erilaisten heurististen hakutoimintojen avulla, mutta ne eivät luonteensa vuoksi takaa optimaalisen ratkaisun löytymistä. Toinen heurististen menetelmien ominaisuus on se, että niiden tuottama sävellys ei välttämättä noudata alkuperäisen Markovin ketjun todennäköisyysjakaumaa, mikä voi sovellusalueesta riippuen olla joko toivottua tai ei-toivottua.

Eräs erimerkki heuristiikan hyödyntämisestä on Davismoonin ja Ecclesin esittelemä menetelmä, jossa alkuperäisen Markovin ketjun tuottama sävellys muokataan vaiheittain mukailemaan ennaltamääritettyjä rajoitteita simuloituun jäähdytykseen perustuvan optimoinnin avulla [8]. Tuloksista havaittiin, että menetelmällä tuotetut melodiat olivat lähes optimaalisia rajoitteiden noudattamisen osalta. Menetelmän havaittiin toimivan erinomaisesti uuden musiikin tuottamiseen sillä edellytyksellä, että tuotetun sävellyksen ei tarvitse noudattaa opetuaineistoa ja rajoitteita absoluuttisesti.

Pachet ym. puolestaan esittelivät ratkaisun, jonka avulla opetusaineisto ja rajoitteet voidaan yhdistää uudeksi rajoitteet toteuttavaksi ketjuksi, joka on tilastollisesti ekvivalentti alkuperäisen ketjun kanssa [7]. Ratkaisu kykenee toteuttamaan sellaiset rajoitteet, joiden laajuus ei ylitä ketjun laajuutta. Yleisessä tapauksessa, kontrollirajoitteiden ollessa mielivaltaisia laajuudeltaan, Markovin ominaisuutta noudattavan ratkaisun tuottaminen ei ole mahdollista. Käytännössä siis esimerkiksi ensimmäisen asteen ketjuun voidaan soveltaa rajoitteita, jotka ovat unäärisiä tai binäärisiä. Tutkimuksessa rajoituttiin seuraaviin unäärisiin rajoitteisiin, joiden avulla tuotettiin syötteen mittaisia sekvenssejä:

- Jatkuvuus: syötteen ja ulostulon loppusävelet identtisiä
- Variaatio: syötteen ja ulostulon alku- ja loppusävelet identtisiä
- Vastaus: syötteen alkusävel ja ulostulon loppusävel identtisiä

Esitelty menetelmä perustuu Markovin ketjun muuttamiseen epähomogeeniseksi Markovin prosessiksi (engl. non-homogenous Markov process, NHMP) [7]. NHMP:tä voidaan ajatella Markovin ketjuna, jonka tilasiirtymämatriisit muuttuvat ajan funktiona. L mittainen NHMP voidaan siis kuvata siirtymämatriisien sarjana $\tilde{M}^{(i)}, i = 0, \ldots, L-1$. Kuvassa 4.1 on havainnollistettu NHMP:n muuttuvia siirtymätodennäköisyyksiä. Valittuamme ensimmäisen tuotettavan merkin alkeistodennäköisyyksien jakaumasta $\tilde{M}^0$, valitsemme seuraavan tuotettavan merkin tilasiirtymämatriisista $\tilde{M}^1$, jonka jälkeen etenemme vastaavalla tavalla aina viimeiseen tilasiirtymämatriisiin $\tilde{M}^3$ saakka.

$$\begin{array}{ccc} A & B & C \end{array}$$
$$\begin{bmatrix} 0.3 & 0.2 & 0.5 \end{bmatrix}$$

(a) $\tilde{M}^0$

$$\begin{array}{c|c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0.1 & 0.6 & 0.3 \\ 0.4 & 0 & 0.6 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

(b) $\tilde{M}^1$

$$\begin{array}{c|c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0.8 & 0.2 & 0 \\ 0 & 0.1 & 0.9 \end{bmatrix} \end{array}$$

(c) $\tilde{M}^2$

$$\begin{array}{c|c} & \begin{array}{ccc} A & B & C \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{array}$$

(d) $\tilde{M}^3$

Kuva 4.1: Tila-avaruudessa $S = \{A, B, C\}$ operoiva epähomogeeninen Markovin prosessi $\tilde{M}$, jolla voidaan tuottaa merkkiin B päättyviä neljän merkin mittaisia merkkijonoja.

Rajoitetun mallin muodostaminen tapahtuu kahdessa vaiheessa [7]. Ensimmäisessä vaiheessa rajoitteiden implisiittisesti tai eksplisiittisesti kieltämät tilasiirtymät poistetaan ja muodostetaan väliaikaiset tilasiirtymämatriisit. Koska välivaiheena rajoitteet muotoillaan rajoitteiden tyydyttämisongelmaksi (engl. constraint satisfaction problem), joka pakotetaan kaarikonsistentiksi (engl. arc-consistency), voidaan tyhjä ratkaisuavaruus havaita ennen uuden mallin viimeistelyä ja hyödyntämistä. Käytännön tasolla ensimmäisen vaiheen jälkeen on muodostettu NHMP:n väliaikaiset tilasiirtymämatriisit $Z^{(i)}, i = 0, \ldots, L-1$. $Z^{(0)}$ vastaa alkuperäisen Markovin ketjun alkeistodennäköisyyksien jakaumaa ja tilasiirtymämatriisit $Z^{(i)}, i = 1, \ldots, L-1$ vastaavat alkuperäistä tilasiirtymämatriisia sillä erotuksella, että rajoitteiden kieltämät tilasiirtymät on poistettu jokaisesta $Z^{(i)}, i = 0, \ldots, L-1$. Ensimmäisen vaiheen tuottamat väliaikaiset tilasiirtymämatriisit eivät siis ole stokastisia, sillä jokaisen todennäköisyysvektorin todennäköisyyksien summa ei ole 1.

Toisessa vaiheessa väliaikaiset tilasiirtymämatriisit uudelleennormalisoidaan, jonka tuloksena muodostuu NHMP $\tilde{M}$ [7]. Aluksi tilasiirtymämatriisi $Z^{(L-1)}$ normalisoidaan erillisenä, samaan tapaan kuin aliluvussa 2.1. Seuraavaksi normalisoinnista aiheutuneet häiriöt propagoidaan takautuvasti aina todennäköisyysvektoriin $Z^{(0)}$ saakka, artikkelissa määritellyn rekursioyhtälön mukaisesti. Normalisoinnin jälkeen NHMP $\tilde{M}$ toteuttaa rajoitteet ja on tilastollisesti ekvivalentti alkuperäisen Markovin ketjun kanssa.

Tutkimusryhmä sovelsi menetelmäänsä kahteen reaaliaikaista toimintaa vaativaan sovellusalueeseen [7]. Ensimmäisessä sovellusalueessa opetusaineiston pohjalta opetettiin Markovin ketju, jonka jälkeen siitä muokattiin valitun rajoitteen ja syötteen pohjalta NHMP, jota hyödynnettiin uusien annetun syötteen mittaisten sekvenssien tuottamiseen. Rajoitteet valittiin aiemmin esitellystä joukosta: jatkuvuus, variaatio ja vastaus. Tuotetuista sekvensseistä havaittiin, että ne edustivat selvästi opetusaineiston musiikillista kategoriaa ja että yksinkertaisilla unäärisillä rajoitteilla voitiin vaikuttaa globaalisti koko tuotettavan sekvenssin rakenteeseen.

Toisen sovellusalueen opetusaineistona toimi jazzkitaristi John McLaughlinin improvisoitu nk. tikapuusoolo, konserttitaltioinnista vuodelta 1977 [7]. Syötteenä käytettiin nousevia jazz-asteikkoja ja rajoitteina käytettiin kolmea kiinnitettyä säveltä alkuperäisestä opetusaineistosta. Kiinnitetyt sävelet olivat opetusaineiston ensimmäinen, keskimmäinen ja viimeinen sävel. Tuloksena havaittiin, että tuotetut sekvenssit olivat nousevien ja laskevien tikapuumelodioiden konkatenaatioita, jotka noudattivat opetusaineiston harmonista kontekstia ja tyydyttivät kolmen kiinnitetyn sävelen rajoitteen. Käytännön tasolla onnistuttiin siis saavuttamaan vaihtelevan kontekstin yksinkertaisia representaatioita, jotka noudattivat sävelasteikon asettamaa tonaalisuutta. Tonaalisuudella tarkoitetaan tässä yhteydessä eri sävelkorkeuksien hierarkisia suhteita sävelasteikon keskeiseen säveleen, toonikaan, nähden.

4.2 Hierarkinen Markov-mallinnus

Eräs Markovin ketjujen ongelma on se, että tuotettu stokastinen malli perustuu näytteenottoon käytettyyn objektiluokkaan, eli analyysin taso on valittu etukäteen [9]. Jos malli perustuu nuottien välisiin siirtymiin, on hyvin mahdollista, että se ei huomioi muiden tasojen rakenteita. Perinteinen Markovin malli on siis yksiulotteinen ja kuvaa tietyn tason siirtymiä hyvin, mutta rikkoo samalla mahdollisesti muiden tasojen rakenteita. Monitasoinen rakenteiden huomiointi on mahdollista hierarkisten Markov-mallien avulla, jotka voidaan perustaa joko Markovin ketjuille tai piilomalleille. Luvussa 3 kuvattiin yksinkertainen tapa huomioida musiikissa ilmenevät tauot hierarkisen mallin avulla, mutta osoittautuu, että laajemmassakin kokonaiskuvassa ilmenee sopivia käyttökohteita hierarkiselle mallinnukselle.

Eräs esimerkki hierarkisuuden hyödyntämisestä on Allanin esittelemä menetelmä syötteenä annettujen sopraanojen harmonisointiin Markovin piilomallien avulla [10]. Piilomallien opetusaineistona käytettiin J.S. Bachin koraaleja, jotka oli segmentoitu hierarkisesti äänialojen perusteella. Harmonisointi jaettiin kolmeen alitehtävään, joista ensimmäisessä rakennettiin sävellyksen harmoninen runko. Syötteenä annettua sopraanoa käsiteltiin piilomallin tuottamana ulostulona ja sen pohjana olevia harmonisia symboleja mallin piilotettuina tiloina, joiden todennäköisin sekvenssi saatiin selville aliluvussa 2.2 kuvattua *Viterbi*-algoritmia hyödyntäen. Samoin meneteltiin toisessa vaiheessa, jossa harmonisten symbolien sekvenssin perusteella tuotettiin harmonisaation sointurunko. Viimeisessä vaiheessa tuotettiin ornamentoidut basso-, altto-, ja tenorimelodiat kolmen toisistaan erillisen piilomallin avulla, niin ikään *Viterbi*-algoritmia hyödyntäen. Tyypillisesti ihmissäveltäjien tuottamat harmonisaatiot pohjautuvat vahvaan traditioon ja yksinkertaisiin selviin sääntöihin. Onkin siksi merkittävää, että Allanin menetelmällä pystytään tuottamaan onnistuneita harmonisointeja automatisoidusti, ilman erillistä aluekohtaista harmonista informaatiota tai ohjausta.

Toisaalta hierarkisia malleja voidaan käyttää myös Thorntonin kuvailemalla tavalla, joka muistuttaa generatiivista kielioppia siltä osin, että sävellyksen tuottamiseen käytetään kontekstiherkille kieliopeille tyypillisiä korvaussääntöjä [9]. Jotkin korvaussäännöistä ovat kuitenkin stokastisia, eli käytännössä osa kontekstiherkän kieliopin muuttujista on korvattu Markovin ketjuilla, joiden avulla johdetaan lisää muuttujia ja päätemerkkejä. Kuvassa 4.2 on kuvattu eräs esimerkki Thorntonin mallin mukaisesta rekursiivisesta, hierarkista mallia hyödyntävästä sävellysprosessista. Tutkimus osoittaa, että esitelty malli yhdistää onnistuneesti hierarkisen joustavuuden ja tilastollisen lähestymistavan, joka ilmenee tuotoksissa rakenteellisena organisointina, jota ei voitaisi muutoin saavuttaa ilman monitasoista analyysiä.

Symbolit	Korvaussäännöt	Todennäköisyydet
S	$S \to A\ B$	
$A\ B$	$A \to D\ C/D$	$P(C\|D) = 0.5$
		$P(D\|D) = 0.5$
	$B \to c\ e/f$	$P(c) = 1.0$
		$P(e\|c) = 0.66$
		$P(f\|c) = 0.34$
$D\ C\ c\ e$	$D \to b/c\ d/e$	$P(b) = 0.5$
		$P(c) = 0.5$
		$P(d\|b) = 0.2$
		$P(e\|b) = 0.8$
		$P(d\|c) = 0.3$
		$P(e\|c) = 0.7$
	$C \to g\ c/d/e$	$P(g) = 1.0$
		$P(c\|g) = 0.2$
		$P(d\|g) = 0.4$
		$P(e\|g) = 0.4$
$b\ e\ g\ d\ c\ e$		

Kuva 4.2: Thorntonin mallin mukainen esimerkki symbolien johtamisesta säveliksi.

4.3 Hahmontunnistus

Hahmontunnistus (engl. pattern recognition) on tieteenala, jonka tavoitteena on objektien luokittelu erilaisiin kategorioihin tai luokkiin [11]. Sovellusalueesta riippuen objektit, yleisesti hahmot, voivat olla esimerkiksi erilaisia kuvia, aaltomuotoisia signaaleja tai tekstikatkelmia. Hahmojentunnistuksen historia ulottuu kauas ja se keskittyi aluksi vahvasti erilaisiin tilastotieteen menetelmiin. Nykyään hahmontunnistus on elintärkeä osa erilaisia koneoppimis- ja tekoälyjärjestelmiä.

Erilaisten hahmontunnistusmenetelmien kirjo on laaja. Musiikillisen materiaalin käsittelyssä vahvimmiksi menetelmiksi ovat osoittautuneet erilaiset tilastotieteeseen pohjautuvat koneoppimismenetelmät, rakenteellinen ja syntaktinen analyysi sekä neuroverkkoihin perustuvat ratkaisut [12]. Koneoppimiskeinoista esimerkkeinä voidaan mainita naiivit Bayes luokittelijat (engl. naive Bayes classifier), lineaariset luokittelijat (engl. linear classifier) ja k-lähimmän naapurin menetelmä (engl. k-nearest neighbour method). Rakenteen ja syntaksin analyysissä puolestaan kieliopillinen analyysi (engl. grammatical inference), kontekstiherkät kieliopit ja erilaiset verkkoesitykset ovat olleet tutkijoiden suosiossa. Neuroverkkojen osalta itseorganisoituvat kartat (engl. self-organizing map) ja perinteiset eteenpäinsyöttävät monikerroksiset perseptroniverkot ovat osoittautuneet toimiviksi (engl. multilayer perceptron).

Musiikillisen materiaalin parissa tehty hahmojentunnistukseen liittyvä tutkimustyö on pääasiassa keskittynyt musiikillisen informaation hakuun (engl. musical information retrieval) [6]. Sen avulla on mahdollistettu esimerkiksi palveluita, jotka yhdistävät hyräillyn tai soitetun melodiakatkelman tiettyyn sävellykseen. Hahmojen poiminta algoritmista sävellystä varten on kuitenkin ollut verrattain harvinaista. Sävellyksissä toistuvien melodioiden ja harmonisten rakenteiden aiheuttama korkea itsesimilaarisuus tekee hahmontunnistuksesta potentiaalisesti hyvin voimakkaan työkalun algoritmisessa sävellyksessä. Luonteva kohde hahmontunnistukselle on opetusaineiston esikäsittely, josta esimerkkeinä voidaan mainita Markovin ketjun tilajoukon tai generatiivisen kieliopin päätemerkkien määrittäminen.

Verbeurgt ym. sovelsivat hahmontunnistusta yhdessä Markovin piilomallien kanssa [6]. Varsinainen tutkijoiden esittämän menetelmän toteutus jakautuu kolmeen osaan. Ensimmäisessä vaiheessa musiikillisen teoksen motiivit poimitaan suffiksipuuhun. Motiivien poiminta tapahtuu kolmiulotteisesti, ulottuvuuksien ollessa yksittäisten nuottien muodostamien hahmojen ajallinen sijoittuminen, relatiivinen sävelkorkeus ja suhteellinen kesto. Toisessa vaiheessa suffiksipuun sisäsolmuissa sijaitsevien toistuvien motiivien välille määritetään opetusaineiston perusteella siirtymätodennäköisyydet, pidempiä kohdemotiiveja painottaen. Jokaiselle motiiville määritetään myös todennäköisyysperustainen emissiofunktio, jonka avulla valinta sävellykseen lisättävän motiivin sävelkorkeudesta ja -kestosta suoritetaan. Menetelmän viimeisessä vaiheessa uusi sävellys tuotetaan piilomallin avulla. Teoksen avaava motiivi valittiin satunnaisesti opetusaineiston avaavien motiivien joukosta, kun taas sulkevat motiivit päätettiin sivuuttaa määrittämällä sävellykset kahdenkymmenen tilasiirtymän mittaisiksi.

Tutkimuksessa havaittiin, että vaikka tuotettu sävellys ei muistuttanutkaan opetusaineiston sävellyksiä, se noudatti kuitenkin pääpiirteissään musiikin teoriaa [6]. Erityisesti sävelet olivat pääasiassa samaan sävelasteikkoon kuuluvia ja siirtymät nuottien välillä suurimmalta osin soljuvia siirtymiä vierekkäisten sävelasteikon nuottien välillä. Poimittujen motiivien käyttö Markovin ketjun solmuissa siis tarjosi rakenteen, suunnan ja jatkuvuuden tuntua tasaisempien siirtymien ansiosta. Tutkimuksen tulosten vertailukohteena käytettiin perinteistä Markovin ketjujen sovellusta, joka pohjautui yksittäisten nuottien ja niiden välisten siirtymäominaisuuksien mallintamiseen. Tällä perinteisemmällä menetelmällä sävelletty musiikkia vaivasi musikaalisen kontekstin puute ja se kärsi ajoittain suurien intervallierojen aiheuttamasta jatkuvuuden heikkenemisestä.

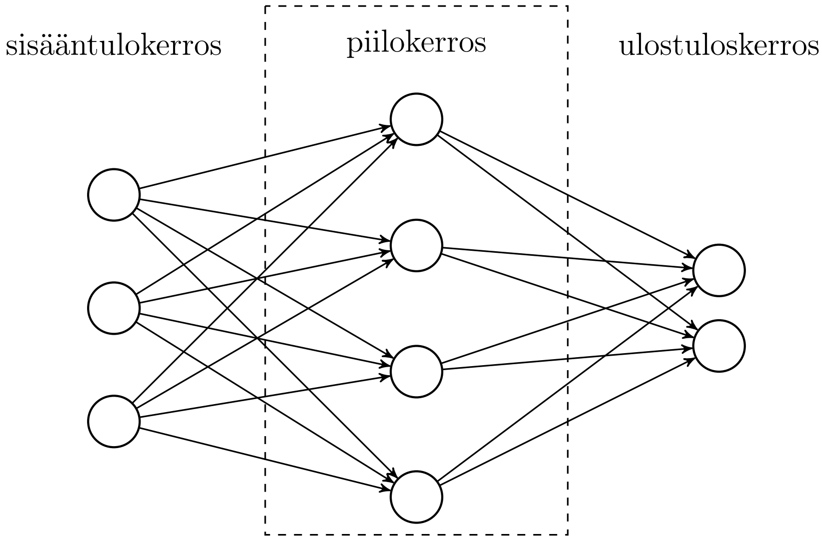
4.4 Neuroverkko

Keinotekoinen neuroverkko (engl. artificial neural network) on tietojenkäsittelymalli, joka kehitettiin alunperin kuvantunnistuksen ja tiedonluokittelun työvälineeksi [1]. Keinotekoinen neuroverkko (myöh. neuroverkko) pohjautuu ihmisten ja useimpien eläinten aivokuoren rakenteeseen ja tapaan käsitellä informaatiota subsymbolisella tasolla.

Neuroverkot koostuvat tasoihin järjestetyistä neuroneista sekä neuronien kynnysarvojen päivitysfunktiosta [1]. Neuroverkon tasot muodostuvat sisään- ja ulostulokerroksista sekä vähintään yhdestä edellisten väliin jäävästä piilokerroksesta. Kuvassa 4.3 on kuvattu edellä mainittua rakennetta yksinkertaisen 3-4-2 -kerrosrakenteisen neuroverkon avulla. Yksittäinen neuroni koostuu sisään- ja ulostuloista sekä propagaatio-, aktivaatio- ja ulostulofunktioista. Sisääntulokerroksen neuroni saa syötteensä suoraan ympäristöstä, kun taas muiden kerrosten neuronit saavat syötteensä edellisen kerroksen neuroneilta. Propagaatiofunktio yhdistää kaikki neuronin saamat syötteet yhdeksi tietueeksi, jonka jälkeen aktivaatiofunktio tuottaa neuronin aktivaatiotilan ennalta määrätyn kynnysarvon ja propagaatiofunktion tuottaman arvon perusteella. Ulostulofunktio muodostaa neuronin ulostulon aktivaatiotilan perusteella. Ulostulokerroksen neuronit antavat ulostulonsa tarkasteltavassa muodossa kun taas muiden kerrosten neuronit antavat ulostulonsa syötteenä niitä seuraavan kerroksen neuroneille.

Vaikka neuroverkkojen tiedonkäsittelykapasiteetti on kovin matala biologiseen malliin verrattuna, pystyvät ne ratkaisemaan korkeankin kompleksisuuden tehtäviä ajallisesti ja laadullisesti tyydyttävin lopputuloksin [1]. Musiikillisen materiaalin moniulotteinen ja laaja kontekstisidonnaisuus aiheuttaa kuitenkin haasteita myös neuroverkoille. Näistä haasteista johtuen osoittautuukin, että vain muutama neuroverkkomalli soveltuu musiikillisen informaation prosessointiin ja niidenkin hyödyntäminen vaatii huolellista neuroverkon rakenteen ja parametrien määrittelyä. Suurin osa ongelmista johtuu siitä, ettei neuroverkkojen kynnysarvojen mukauttamiseen käytetty vastavirta-algoritmi (engl. backpropagation algorithm) ja sen muunnelmat kykene käsittelemään musiikillista kontekstia tyhjentävästi. Toisaalta neuroverkkojen taipumus pelkistyä muuttumattomiksi rakenteiksi ei myöskään tue monimuotoisten pitkien musiikillisten sekvenssien tuottamista. Huomionarvoinen seikka on myös neuroverkkojen opettamiseen vaadittava verrattain pitkä aika, jota pidempien musiikillisten sekvenssien tuottaminen vaatii. Voimakkaasti kontekstisidonnaiseen tai alueelliseen tietoon pohjautuvien musiikillisten tuotosten tuottamiseen generatiiviset kieliopit ja sääntöpohjaiset järjestelmät toimivat siis merkittävästi neuroverkkoja paremmin.

Neuroverkoilla on kuitenkin selkeä paikkansa musiikillisen materiaalin tuottamisessa [1]. Erityisesti lyhyiden musiikillisten sekvenssien tuottamiseen neuroverkot toimivat Markovin ketjuja ja generatiivisia kielioppeja paremmin, sillä niiden avulla voidaan tuottaa yllättäviä motiiveja, jotka eivät esiinny opetusaineistossa, mutta täyttävät silti sen antamat implisiittiset vaatimukset ja rajoitteet. Siten neuroverkot motivoivat niiden soveltamiseen innovatiivisten sävellyksellisten konseptien kanssa. Toisaalta neuroverkot soveltuvat myös erinomaisesti täydentämään muita algoritmisen sävellyksen menetelmiä erilaisissa hybridimenetelmissä.



Kuva 4.3: Yksinkertaisen neuroverkon esitys verkkona, jossa solmut kuvaavat neuroneita ja kaaret solmujen välillä neuronien välisiä yhteyksiä. Sisääntulokerros koostuu kolmesta neuronista, piilokerros neljästä neuronista ja ulostulokerros kahdesta neuronista.

Verbeurgt ym. ehostivat aliluvussa 4.3 esiteltyä menetelmää myöhemmin neuroverkon avulla [13], joka korvasi todennäköisyysperustaiset emissiofunktiot. Emissiofunktioiden heikkoudeksi tulkittiin se, että ne riippuivat vain nykyisestä tilasta, eivätkä huomioineet lainkaan edellisen motiivin sävelkorkeutta ja -kestoa. Neuroverkon tehtävä oli mallintaa nykyisen tilan pohjasävelen ja -keston ehdollista todennäköisyysjakaumaa kun se saa syötteekseen edellisen tilan pohjasävelen ja -keston. Neuroverkon opetuksessa käytettiin tavallista vastavirta-algoritmia ja opetusaineistosta satunnaisesti valittuja pisteitä sekä näistä pisteistä haarautuvaa suffiksipuun motiivia. Koska suffiksipuun juuren ja lehtisolmun välillä on mahdollisesti useita haarautumia, valittiin yksi niistä todennäköisyyksiin perustuen, jälleen pidempiä motiiveja painottaen.

Neuroverkon avulla tuotetuilla emissioilla tuloksista saatiin vielä vakuuttavampia kuin todennäköisyyksiin perustuvilla emissiofunktioilla [13]. Tuloksista havaittiin, että opetusaineistossa toistuva teema, päämotiivi, toistui useita kertoja sävellyksessä. Menetelmä onnistui siis erinomaisesti opetusaineiston pääpiirteiden poiminnassa ja käytössä. Sävellykset olivat edellisten tulosten tapaan sävelasteikolleen uskollisia ja niiden sisältämät siirtymät olivat tasaisia. Poikkeuksellisesti havaittiin kuitenkin, että myös suuria intervallieroja sisältävät siirtymät olivat sujuvia.

4.5 Generatiivinen kielioppi

Generatiiviset kieliopit (engl. generative grammar) ovat keskeinen osa musiikillista analyysiä algoritmisessa säveltämisessä [1]. Ne mahdollistavat musiikillisen materiaalin kontekstisidonnaisen ja hierarkisen organisoinnin. Uusien ilmaisujen (engl. expression) tuottaminen generatiivisen kieliopin avulla perustuu korvaussääntöihin (engl. rewriting rules), joissa lausekkeen vasemmanpuoleiset symbolit korvataan lausekkeen oikeanpuoleisilla symboleilla. Symbolit voidaan jakaa muuttujiin (engl. variable, non-terminal symbol) ja päätemerkkeihin (engl. terminal symbol). Symbolisekvenssin tuottaminen alkaa erityisestä lähtösymbolista, jonka jälkeen korvaussääntöjä sovelletaan niin kauan, että kaikki muuttujat on korvattu päätemerkeillä. Korvaussäännöt siis muodostavat hierarkisen rakenteen, jonka avulla olemassaolevan ilmaisun oikeellisuus voidaan tarkistaa tai uusien ilmaisujen tuottaminen on mahdollista.

Generatiivisten kielioppien perusluonne on kuitenkin rajoittava musiikillisessa kontekstissa, sillä niiden perustana on peräkkäiseen käsittelyyn perustuva malli, mikä jättää musiikissa samanaikaisesti tapahtuvat ilmiöt, moniäänisyyden, huomiotta [1]. Tämän vuoksi niitä käytetäänkin lähinnä sellaisten musiikillisten tyylien mallintamiseen, jotka mahdollistavat musiikkimateriaalin peräkkäisen käsittelyn. Esimerkkejä sopivista tyyleistä ovat muunmuassa eurooppalainen taidemusiikki yksiääniset melodiat ja jazz-musiikin harmoniset sointukulut. Huomionarvoista on myös se, että kielioppien yhteydessä käytetyt käsitteet *syvä rakenne* ja *semantiikka* (merkitys) eivät sovellu musiikilliseen materiaaliin. Siinä missä kielitieteellisen ilmaisun rakenneosat voivat esiintyä eri positioissa semantiikan muuttumatta, musiikillisen materiaalin uudelleen järjestely lähes poikkeuksetta johtaa erilaiseen musiikilliseen informaatioon. Lisäksi semanttisten menetelmien käyttö johtaa ongelmiin, sillä merkityksen löytäminen musiikillisen rakenteen oikeasta esitystavasta huolimatta ei ole mahdollista.

Generatiivisten kielioppien vahvuudet ovat niiden korkeassa tuottoasteessa ja mahdollisuudessa kompleksisten musiikillisten rakenteiden tuottamiseen verrattain pienellä määrällä korvaussääntöjä [1]. Kieliopillinen analyysi (engl. grammatical inference) puolestaan tarjoaa mielenkiintoisia vaihtoehtoja tyylien imitointiin muuttuvan kontekstisyvyyden käsittelyn mahdollisuuden ansiosta. Kokonaisuudessaan generoivat kieliopit muodostavat hyödyllisen algoritmien joukon musiikilliseen analyysiin, tyylin imitointiin ja jopa aitoon säveltämiseen.

Gillick ym. hyödynsivät generatiivista kielioppia yhdistettynä Markovin ketjuun tutkimuksessa, jonka tavoitteena oli jazz -soolojen tuottaminen [14]. Heidän päämääränään oli erityisesti opetusaineiston tyyliä imitoivien soolojen tuottaminen. Menetelmän ensimmäisessä vaiheessa melodiat abstraktoitiin jakamalla ne luisuihin (engl. slopes), joita yhdistelemällä melodiset muodot (engl. countour), melodiafragmentit, rakentuvat. Seuraavaksi melodiafragmenteista muodostettiin kielioppi kieliopillisen analyysin avulla. Kolmannessa vaiheessa melodiafragmentit ryvästettiin seitsemään parametriin pohjautuvan euklidisen etäisyyden perusteella. Lopulta näiden melodiafragmenttien ryppäiden välille määritettiin siirtymätodennäköisyydet, niin että lopputuloksena oli todennäköisyysperustainen kielioppi, jossa Markovin ketju on ikään kuin sisäänrakennettuna.

Tutkimuksen tuloksia arvioitiin sekä laadullisesti että kokeellisesti [14]. Lyhyet soolot olivat säännönmukaisesti laadultaan hyviä ja erehdyttäviä niiden lähteen suhteen, sillä niiden arvioitiin muistuttavan kykenevän solistin improvisaatiota tai olevan vakuuttavia imitaatioita. Pidemmät soolot kärsivät melodian vaeltelun aiheuttamasta suunnan ja rakenteen puutteesta. Tutkimuksessa päädyttiin käyttämään neljännen asteen Markovin ketjuja, sillä sekä matalamman että korkeamman asteen ketjujen tuotoksien havaittiin olevan merkittävästi heikompia. Korkeampaa astetta olevien ketjujen arveltiin kuitenkin tuottavan parempia tuloksia, mikäli opetusaineiston kokoa kasvatettaisiin merkittävästi. Ihmiskokeessa pyrittiin yhdistämään algoritmisesti tuotettu soolo tiettyyn solistiin. Ihmiskokeen tulokset olivat hyvät, sillä tuotetut soolot kyettiin yhdistämään oikeaan henkilöön yli 85% varmuudella. Lisäksi imitaatioiden arvioitiin olevan pääsääntöisesti “jokseenkin lähellä” tai “melko lähellä” kyseisen solistin tyyliä.

4.6 Geneettinen algoritmi

Geneettiset algoritmit (engl. genetic algorithm) kuuluvat evoluutioalgoritmien (engl. evolutionary algorithm) joukkoon, joiden perimmäinen tarkoitus on mallintaa evoluutioprosesseja tietokoneella suoritettavissa simulaatioissa [1]. Ne sopivat erityisen hyvin ongelma-alueisiin, joita on vaikea mallintaa matemaattisesti tai joille ei ole olemassa eksplisiittisesti määriteltyä sääntöjoukkoa, sillä aluekohtainen tieto ongelmasta ei ole välttämätöntä niiden toiminnan kannalta.

Biologiseen malliin verrattuna tietokoneohjelma toimii elinympäristönä, joka tarjoaa tietyt olosuhteet selviytymiselle ja periytymiselle [1]. Keinotekoisessa elinympäristössä tuotetaan yksilöistä koostuvia populaatioita, joiden mukautumista asetettuun tavoitteeseen arvioidaan sopeutuvuusfunktion (engl. fitness function) avulla. Uusia populaatioita tuotetaan evoluutio-operaattoreiden, useimmiten rekombinaation ja mutaation avulla. Sekä rekombinaatio että mutaatio voivat tapahtua monin eri tavoin. Rekombinaatiossa vanhempina toimivien yksilöiden vastaavat osat vaihdetaan keskenään ja mutaatiossa osa yksilöstä korvataan stokastisin menetelmin. Uusi populaation sukupolvi koostuu sopivimpien yksilöiden jälkeläisistä, joista merkittävä osa tuotetaan rekombinaation keinoin. Pieni osa mutaation avulla tuotettuja jälkeläisiä varmistaa ettei populaation sopivuus supistu paikalliseen maksimiin. Usein myös osa kaikkein sopivimmista yksilöistä jatkaa suoraan seuraavaan populaatioon. Uusien populaatioiden tuottamista jatketaan niin kauan, kunnes tavoitteen tarpeeksi hyvin tai täysin täyttävä yksilö on löytynyt.

Algoritmisen säveltämisen osalta geneettisten algoritmien suurimmat heikkoudet ilmenevät voimakkaasti kontekstiriippuvaisten tehtävien kohdalla [1]. Esimerkiksi sointuprogressioiden tuottaminen on käytännössä lähes mahdotonta, sillä yksilön sopivuuden parantuessa tietyssä ulottuvuudessa sopivuus samanaikaisesti heikkenee muissa ulottuvuuksissa. Toinen heikkous on geneettisten algoritmien taipumus tuottaa hyviä tuloksia pienissä sävellystehtävissä, kun kokonaisten järkeenkäypien sävellysten tuottaminen on haastavaa. Kolmas huomioitava seikka on se, että täysin sopivan yksilön löytymistä ei voida taata joka suorituskerralla, sillä tila-avaruuteen kohdistuva haku on heuristinen täydellisen sijaan.

Geneettisten algoritmien yhteydessä Markovin malleja voidaan hyödyntää sekä populaation yksilöiden sopivuuden arvioinnissa että uuden populaation tuottamisessa. Lo ym. tutkivat tasoitettujen n-grammien hyödyntämistä geneettisen algoritmin sopivuusfunktiona [15]. Tutkimuksessa suoritettiin neljä koetta, joiden tavoitteena oli selvittää millaisia ominaisuuksia n-grammeilla on sopivuusfunktiona ja millaiset evoluutioalgoritmit sopivat optimoimaan populaatiota ennalta asetettua tavoitetta kohti. Ensimmäiset kaksi koetta osoittivat, että n-grammeilla oli kyky luokitella saman tyylilajin eri säveltäjien teoksia oikein yli 80% todennäköisyydellä. Toinen koe osoitti myös, että tuotetut melodiat eivät olleet mielekkäimpiä varsinaisessa tavoitekohdassa, vaan mielenkiintoisimmat melodiat sijaitsivat karkeasti n-grammin kaikkein todennäköisimmän sekvenssin (engl. maximum likelihood sequence, MLS) ja täysin satunnaisen sekvenssin välillä, koska MLS -sekvenssien melodiat olivat liian repetitiivisiä. Kolmannessa kokeessa testattiin yhdeksän eri evoluutio-operaattorin toimintaa ja havaittiin ettei yhdelläkään kyetty kehittämään melodioita lähellekään MLS -sekvenssiä. Kolmannen kokeen tuloksesta voitiin siten päätellä, että mahdolliset ongelmat eivät niinkään piile n-grammin käytössä sopivuusfunktiona vaan pikemminkin evoluutio-operaattoreiden kyvyttömyydessä supeta kohti optimia tulosta. Neljännessä kokeessa puolestaan havaittiin, että käyttämällä relatiiviseen sävelkorkeuteen pikemminkin kuin absoluuttiseen sävelkorkeuteen perustuvia n-grammeja, saavutettiin merkittävästi parempia tuloksia. Yhteenvetona voidaan todeta, että n-grammit toimivat varsin tehokkaasti geneettisen algoritmin sopivuusfunktiona, mutta kiinnostavien melodioiden tuottaminen vaatii evoluutio-operaattoreilta yleistä musiikillista tietoutta.

Manaris ym. puolestaan sovelsivat Markovin ketjuja interaktiivisessa *Monterey Mirror* -sovelluksessa, jonka tehtävä oli tuottaa käyttäjän syötteenä antaman melodian kaltaisia musiikkisekvenssejä [16]. Käyttäjän antaman syötteen perusteellla opetetaan yksittäisten nuottien välisiin siirtymäominaisuuksiin pohjautuva Markovin ketju. Markovin ketjulla tuotetaan stokastisesti alkuperäinen populaatio, jota kehitetään rekombinaation ja mutaation avulla. Rekombinaatiossa kaksi sekvenssiä risteytetään siten, että risteymät ovat konsistentteja Markovin ketjun kanssa. Mutaatiossa satunnainen osa sekvenssistä tuotetaan uusiksi Markovin ketjun avulla. Kun populaation generaatioita on tuotettu ennaltamääritetty määrä, sovellus palauttaa käyttäjälle musiikkisekvenssin, jolla on paras sopivuus hänen antamansa syötteen kanssa olematta kuitenkaan identtinen. Tutkimuksessa havaittiin Markovin ketjujen sopivan hyvin yhteen perinteisten evoluutio-operaattoreiden kanssa niiden ennustettavuuden ansiosta. Niiden rajattu tila-avaruus mahdollistaa reaaliaikaisuutta vaativan toiminnan ja pitkän aikavälin riippuvuuksien toteuttamisen tuotetuissa sekvensseissä. Sovelluksen havaittiin kykenevän vuorovaikutukseen käyttäjänsä musiikillisen luovuuden kanssa ja vastaamaan älykkäillä ja merkityksellisillä sekvensseillä, jotka pohjautuivat käyttäjän syötteen variaatioihin. Siten sovellus nähtiin voimakkaana työkaluna nykyaikaisen musiikin luomiseen sekä sävellysympäristön että meta-instrumenttin asemassa.

5 Johtopäätökset

Algoritmisen säveltämisen kontekstissa on selvää, että Markovin ketjut yksinään eivät ole riittävä ratkaisu uuden musiikillisen materiaalin tuottamiseen. Niiden rakenne mahdollistaa vain perättäisten symbolien kontekstisidonnaisten riippuvuuksien kuvaamisen, ja ne ovat sopivia lähinnä yksiulotteisten symbolisekvenssien mallintamiseen, mikä ei olekaan yllätys niiden lingvistinen tausta huomioiden. Nämä rakenteelliset ominaisuudet eivät ole omiaan kuvaamaan musiikillista informaatiota, mikä pääsääntöisesti koostuu vertikaalisesta ja kerrostetusta informaatiosta (polyfonia eli moniäänisyys) [1]. Myöskään korkeampiasteisten ketjujen taipumus toistaa informaatiota identtisenä sekä matalampiasteisten ketjujen liika satunnaisuus eivät ole tavoiteltavia ominaisuuksia musiikillisen materiaalin tuottamisessa [5].

Näistä rajoitteista huolimatta Markovin ketjut vaikuttavat soveltuvan erinomaisesti toimimaan yhdessä muiden menetelmien kanssa erilaisissa algoritmisen säveltämisen hybridiratkaisuissa niiden tehokkaan ohjelmoinnin ansiosta. Fernández ja Vico kartoituksessaan algoritmisen säveltämisen tekoälymenetelmistä [5] ovat niin ikään osoittaneet Markovin ketjujen olevan erittäin käyttökelpoisia ja yleisiä algoritmisen säveltämisen hybridimenetelmissä. Lisäksi luvun 4 esimerkkien valossa on selvää, että Markovin mallien käyttö algoritmisessa sävellystyössä hyötyy erityisesti siitä, että sävellystä ei yritetä tehdä yksittäisten nuottien tasolla. Tällöin lähes väistämättä tarvitaan muiden menetelmien tuottamaa lisäinformaatiota.

On hyvin todennäköistä, että algoritmisen säveltämisen aikaansaannoksissa tullaan näkemään puutteita vielä pitkään, varsinkin jos vertailu tapahtuu ihmissäveltäjien tuotoksiin. Algoritmisilla sävellyskeinoilla on kuitenkin suuri potentiaali toimia tehokkaina ideamoottoreina, jotka kykenevät yllättämään keinoillaan luoda uutta satunnaisesta. Toisaalta autonomiseen säveltämiseen liittyvät jatkuvasti etenevät tutkimustulokset osoittavat positiivista suuntaa automatisoidun luovuuden saavuttamisessa.

Lähteet

1 G. Nierhaus, *Algorithmic composition: paradigms of automated music generation*, ch. 1–4, 7, 9. Springer Science & Business Media, 2009.

2 E. Miranda, *Composing music with computers*, ch. 3. Focal Press, 2001.

3 C. Ames, "The markov process as a compositional model: a survey and tutorial," *Leonardo*, vol. 22, no. 2, pp. 175–187, 1989.

4 N. Privault, *Understanding Markov Chains*, ch. 1–4. Springer, 2013.

5 J. D. Fernández and F. Vico, "Ai methods in algorithmic composition: A comprehensive survey," *Journal of Artificial Intelligence Research*, vol. 48, pp. 513–582, 2013.

6 K. Verbeurgt, M. Dinolfo, and M. Fayer, "Extracting patterns in music for composition via markov chains," in *Innovations in Applied Artificial Intelligence* (B. Orchard, C. Yang, and M. Ali, eds.), (Berlin, Heidelberg), pp. 1123–1132, Springer Berlin Heidelberg, 2004.

7 F. Pachet, P. Roy, and G. Barbieri, "Finite-length markov processes with constraints," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

8 S. Davismoon and J. Eccles, "Combining musical constraints with markov transition probabilities to improve the generation of creative musical structures," in *European Conference on the Applications of Evolutionary Computation*, pp. 361–370, Springer, 2010.

9 C. Thornton, "Hierarchical markov modeling for generative music," in *The International Computer Music Conference*, 2009.

10 M. Allan, "Harmonising chorales in the style of johann sebastian bach," Master's thesis, University of Edinburgh, 2002.

11 S. Theodoridis, *Pattern recognition*, ch. 1. Amsterdam ; London: Elsevier/Academic Press, 4th ed ed., 2009. Previous ed.: Amsterdam: Academic, 2003.

12 P. J. P. De Leon and J. M. Inesta, "Pattern recognition approach for music style identification using shallow statistical descriptors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 2, pp. 248–257, 2007.

13 K. Verbeurgt, M. Fayer, and M. Dinolfo, "A hybrid neural-markov approach for learning to compose music by example," in *Advances in Artificial Intelligence* (A. Y. Tawfik and S. D. Goodwin, eds.), (Berlin, Heidelberg), pp. 480–484, Springer Berlin Heidelberg, 2004.

14 J. Gillick, K. Tang, and R. M. Keller, “Machine learning of jazz grammars,” *Computer Music Journal*, vol. 34, no. 3, pp. 56–66, 2010.

15 M. Lo and S. M. Lucas, “Evolving musical sequences with n-gram based trainable fitness functions,” in *2006 IEEE International Conference on Evolutionary Computation*, pp. 601–608, IEEE, 2006.

16 B. Manaris, D. Hughes, and Y. Vassilandonakis, “Monterey mirror: combining markov models, genetic algorithms, and power laws,” in *Proceedings of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011)*, pp. 33–40, Citeseer, 2011.

Liite 1. J.S. Bach - Air

Air

From Orchestral Suite No. 3
BWV 1068

Johann Sebastian Bach
(1685-1750)

Violin I

Edited by
Miguel Angel Cardozo-Montilla

CARACAS, 2017

MCM.1020B.00002.VL1