

Alexi Tarvainen

**Markovin ketjut ja Markovin piilomallit algoritmisessa
säveltämisessä**

Tietotekniikan kandidaatintutkielma

30. huhtikuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Aleksi Tarvainen

Yhteystiedot: a.aleksi.tarvainen@student.jyu.fi

Ohjaaja: Antti-Jussi Lakanen

Työn nimi: Markovin ketjut ja Markovin piilomallit algoritmisessa säveltämisessä

Title in English: Markov chains and hidden Markov models in algorithmic composing

Työ: Kandidaatintutkielma

Sivumäärä: 21+0

Tiivistelmä: Tässä tutkielmassa kartoitettiin Markovin ketjujen ja Markovin piilomallien hyödyntämistä algoritmisessa säveltämisessä. Markovin ketjuja on hyödynnetty algoritmisessa säveltämisessä jo 1950-luvulta lähtien. Kirjallisuuskatsauksen tuloksena selvisi, että menetelmiä käytetään ensisijaisesti musiikkityylien jäljittelyyn. Menetelmien heikkoudeksi nähtiin niiden tuottamien sävellysten yleinen musiikillisen rakenteen puute erityisesti pidemmissä tuotoksissa. Markovin ketjujen etuna oli yksinkertainen ja tehokas toteutus.

Avainsanat: tietokonemusiikki, algoritmisen säveltäminen, Markovin ketjut, Markovin piilomalli

Abstract: This thesis is a survey of Markov chains and Hidden Markov in algorithmic composing. The former have been used in algorithmic composing since the 1950s and the latter are derived from Markov chains. Markov chains and hidden Markov models are primarily used in imitation of musical styles. Pieces generated with the said methods lack musical structure, especially in the long-term. Markov chains' advantage was their simple and efficient implementation.

Keywords: computer music, algorithmic composition, Markov chain, hidden Markov model

Kuviot

Kuvio 1. Markovin ketju esitettyinä tilakaaviona ja matriisina, mukailten Nierhaus (2009, s. 68)	6
Kuvio 2. Markovin piilomalli (Terencehonles 2009)	8

Sisältö

1	JOHDANTO	1
2	ALGORITMINEN SÄVELTÄMINEN	2
2.1	Musiikki ja säveltäminen	2
2.2	Algoritminen säveltäminen	2
3	MARKOVIN KETJUT JA MARKOVIN PIILOMALLI	5
3.1	Markovin ketjut.....	5
3.2	Markovin piilomalli	7
3.3	Markovin ketjut algoritmisessa säveltämisessä	8
3.4	Markovin piilomallit algoritmisessa säveltämisessä.....	10
4	YHTEENVETO.....	13
	LÄHTEET	14

1 Johdanto

Musiikki on osa jokapäiväistä elämäämme ja uudelle musiikille on jatkuva tarve, vaikka länsimaiselle musiikille tyypillisiä läpisävellettyjä sävelmiä on tuotettu jo satojen vuosien ajan. Algoritminen säveltämisen avulla musiikkia voidaan tuottaa ohjelmallisesti. Taustamusiikille, jota voisi soittaa jatkuvalla syötöllä ilman tekijänoikeusmaksuja, on tarvetta muun muassa julkisissa tiloissa, hisseissä ja puhelinpalveluiden odotuslinjoilla (Nierhaus 2009, s. 262). Hendrikx ym. (2013) esittävät, että algoritmisen säveltämistä voisi hyödyntää myös peleissä, jotka tarvitsevat alati uutta sisältöä.

Edwards (2011) väittää, että musiikin säveltäminen on aina ollut kytköksissä säveltäjän omaan ohjelmointiajatteluun (computational thinking) ja algoritmiset sävellyskeinot olisivat vähemmän alttiita toistamaan perinteisempien sävellysmenetelmien kliseitä. Algoritminen säveltäminen sinänsä ei ole uusi idea. Jo 1700-luvulta on peräisin noppapelejä, joissa pelaaja pystyi yhdistelemään pienistä valmiista sävelmien palasista uusia sävellyksiä, kuten valsseja tai poloneeseja (Hedges 1978). Lejaren Hillerin ja Leonard Isaacsonin jousikvartetille tarkoitettua *Illiace suitea* vuodelta 1956 pidetään ensimmäisenä tietokoneella sävellettyinä teoksena (Edwards 2011). Edwards (2011) on sitä mieltä, että tietokoneiden käyttäminen musiikin säveltämiseen on loogista, sillä niistä on tullut merkittävä työväline muillakin taiteenaloilla, kuten esimerkiksi kuvataiteissa.

Tässä kirjallisuuskatsauksessa tarkastellaan Markovin ketjuja ja Markovin piilomalleja algoritmisen säveltämisen kontekstissa. Menetelmien soveltuvuutta algoritmisen säveltämisen eri tehtäviin arvioidaan. Markovin ketjut ovat suhteellisen yksinkertainen malli ja Markovin piilomallit rakentuvat niiden päälle. Vaikka algoritminen säveltäminen ei määritelmän mukaan vaadi tietokoneen käyttämistä, tullaan tässä tutkielmassa keskittymään ensisijaisesti niiden hyödyntämiseen.

Tutkielman olennaisimmat käsitteet esitellään luvussa 2. Luvussa 3 esitellään Markovin ketjut ja Markovin piilomalli sekä perehdytään niiden käyttöön algoritmisessa säveltämisessä. Luku 4 on yhteenveto tutkimuksesta.

2 Algoritminen säveltäminen

Tässä luvussa esitellään tutkielman kannalta keskeiset käsitteet, joita ovat musiikki, säveltäminen ja algoritminen säveltäminen. Lisäksi luodaan yleiskatsaus algoritmisen säveltämisen kenttään ja sen menetelmiin.

2.1 Musiikki ja säveltäminen

Musiikkia voidaan pitää järjestyneinä ääнинä, mutta Kania (2017) huomauttaa, että määritelmä on myös liian laaja, sillä esimerkiksi ihmisten puhe ja monet muut äänilähteet noudattavat tiettyjä sääntöjä. Kania (2017) lainaa aiempaa teosta (Gracyk ja Kania 2011), jossa musiikki määritellään siten, että musiikki on mikä tahansa tapahtuma, joka on tarkoitettu kuultavaksi ja jolla on joitakin musikaalisia piirteitä, kuten rytmi tai sävelkorkeus, tai joka on tarkoitettu kuunneltavaksi sellaisten ominaispiirteiden vuoksi. Musiikkiteoksia voi tallentaa monissa muodoissa, kuten esimerkiksi nuotteina, äänitallenteena tai ohjelmakoodina.

Kotimaisten kielten keskus (2020) määrittelee säveltämisen varsinaisessa merkityksessään seuraavasti: “luoda musiikkia, musiikkiteoksia”. Säveltämisen tuloksena on sävellys eli säveltäjän määrittelemät ohjeet musiikkiteoksen esittämiseksi kuultavana äänenä (Mazzola, Park ja Thalmann 2011).

2.2 Algoritminen säveltäminen

Algoritminen säveltäminen on musiikin säveltämistä formaalisti määriteltyjen sääntöjen avulla (Nierhaus 2009, s. 1). Tavallisesti algoritmeja käytetään ongelmien ratkaisemiseen. Uuden musiikin tuottaminen on kuitenkin ongelma, jolle on vaikea määritellä ennalta mitään lopputilaa, jossa ongelma olisi lopullisesti ratkennut.

Toisinaan algoritmisesta säveltämisestä tai paremminkin sen tuotoksista puhutaan myös generatiivisena musiikkina ja kirjallisuudessakaan harvoin tehdään eroa näiden kahden käsitteen välille. Joitakin pyrkimyksiä käsitteiden erottamiseen on kuitenkin ollut. Wooller ym. (2005) esittelevät algoritmisen säveltämisen menetelmien arvioimiseen viitekehyksen,

jossa yksi akseli on analyttinen-muunnoksellinen-generatiivinen (analytic-transformational-generative). Siinä generatiivisuus tarkoittaa sitä, että generatiivisen algoritmin tuottamaa dataa on enemmän kuin sille on syötetty ja se on musikaalisilta ominaisuuksiltaan laajempaa (Wooller ym. 2005). Collins (2008) puolestaan näkee eron siten, että generatiivinen musiikki viittaa nimenomaan reaaliaikaisiin algoritmisen säveltämisen menetelmiin.

Algoritmiseen säveltämiseen on kehitetty useita menetelmiä. Menetelmät on ollut tapana jaotella deterministisiin ja stokastisiin, vaikkakin nykyisillä menetelmillä on tapana olla sekoituksia molemmista (Edwards 2011). Deterministiset menetelmät tuottavat joka kerta samanlaisen lopputuloksen annetulla syötteellä. Esimerkiksi Lindenmeyer-systeemit (L-systeemit), kieliopit ja soluautomaatit ovat deterministisiä menetelmiä. Stokastiset menetelmät puolestaan hyödyntävät satunnaisuutta. Markovin ketjut ja Markovin piilomallit, kuuluvat Edwardsin (2011) luokittelun mukaan stokastisiin menetelmiin, sillä ne todennäköisesti tuottavat joka kerta eri tuloksen samalla syötteellä.

Muitakin jaottelutapoja on esitetty. Esimerkiksi Liu ja Ting (2017) tekevät jaon kolmeen kategoriaan: matemaattisiin malleihin, kielioppeihin (engl. grammars) sekä laskennalliseen älyyn (computational intelligence). Tässä jaottelussa Markovin ketjut sijoittuvat matemaattisiin malleihin ja Markovin piilomallit laskennalliseen älyyn.

Algoritmista säveltämisestä on erotettavissa kolmenlaisia tavoitteita. Ensimmäiseksi on järjestelmiä, joilla pyritään imitoimaan aiempia musiikkityylejä. Tämä tapahtuu yleensä aiemmasta aineistosta johtamalla. Toisaalta on myös menetelmiä, joilla on pyrkimys tuottaa uudenlaista musiikkia. Viimeiseksi on kehitetty järjestelmiä, jotka kykenevät soittamaan ihmisen rinnalla mukautuen reaaliajassa tämän soitantaan, kuten esimerkiksi MySong (Simon, Morris ja Basu 2008) ja GenJam (Biles 1994). Ensimmäiseksi mainittu tekee automaattisesti säestyksen laulumelodialle ja jälkimmäinen kykenee luomaan improvisoitua jazzsooloa vuorotellen soittajan kanssa.

Algoritmista säveltämistä saatetaan vastustaa, koska ajatellaan, että tietokone säveltää ihmisen puolesta. Tästä vastustuksesta kertoo esimerkiksi se, että tämänkin tutkielman johdannossa mainitusta Lejaren Hilleristä kertova artikkeli sisällytettiin Grove Dictionary of Music and Musicians -teokseen vasta hänen kuolemansa jälkeen. Kuitenkin, jotta ohjelmoitu sävel-

lysprosessi voi edes lähtää käyntiin, on ensin suunniteltava ja toteutettava sävellyksen generoiva järjestelmä. Näiden järjestelmien toteuttaminen vaatii muun muassa suunnittelua ja ohjelmointikielen osaamista. Mikään oikotie musikaalisiin tuotoksiin algoritmisen säveltämisen ei siis ole. (Edwards 2011) Tietokoneen sävellystä ei tarvitse myöskään pitää lopullisena sävellyksenä vaan se voi toimia alkupisteenä tai pelkkänä inspiraation lähteenä.

3 Markovin ketjut ja Markovin piilomalli

Tässä luvussa esitellään ensin Markovin ketjut ja Markovin piilomallit ja sen jälkeen niiden käyttöä algoritmisen musiikin säveltämisessä. Kyseiset menetelmät valikoituivat tutkielman kohteiksi, koska molemmissa satunnaisuus on osa prosessia. Markovin piilomallit hyödyntävät Markovin ketjuja ja ovat siten luonnollinen lisä tutkielmaan.

3.1 Markovin ketjut

Markovin ketjut ovat stokastisia eli ajassa sattumanvaraisesti eteneviä prosesseja, joille pätee

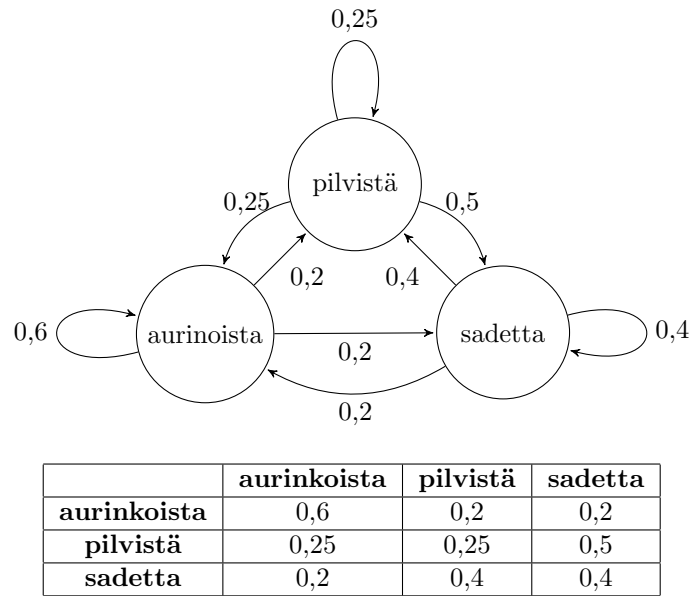
$$P(q_t | q_{t-1}, q_{t-2}, \dots, q_1) = P(q_t | q_{t-1})$$

missä q_1, \dots, q_t ovat ketjun tilajoukon alkioita (Merwe ja Schulze 2011). Toisin sanoen todennäköisyys siirtyä tilaan q_t riippuu ainoastaan nykyisestä tilasta q_{t-1} . Musikaalisessa kontekstissa tilojen määrä t voidaan olettaa äärelliseksi (Ames 1989). Markovin ketjut on nimetty venäläisen matemaatikon Andrei Markovin (1856-1922) mukaan (Nierhaus 2009, s. 67).

Markovin ketjuja on mahdollista esittää tilakaaviona tai matriisina kuten kuviossa 1, jossa molemmat esitystavat kuvaavat samaa Markovin ketjua. Kustakin tilasta lähtevien tilasiirtymien todennäköisyyksien summan on oltava 1. Matriisella tämä on helposti laskettavissa summaamalla rivien alkioit yhteen.

Tilasiirtymämatriisi voidaan muodostaa joko käsin asettamalla kullekin siirtymälle toivottu todennäköisyys tai analysoimalla aineistoa (korpus), josta saadaan muodostettua matriisin tilasiirtymät (Gagniuc 2017, s. 25). Aineistona voi toimia esimerkiksi jokin kokoelma tekstejä, jos halutaan selvittää, miten suurella todennäköisyydellä aakkoset esiintyvät perättäin. Kirjallisuudessa (Nierhaus 2009; Gagniuc 2017) Markovin ketjuja käsiteltäessä usein esiintyvä esimerkki on säähavainnot ja tulevan sään ennustaminen.

Tilasiirtymien todennäköisyydet saadaan laskemalla kunkin tilasiirtymän esiintymismäärä symbolijonossa ja jakamalla se tilasiirtymän lähtötilan esiintymien lukumäärällä (Gagniuc 2017, s. 26). Yhtä tilasiirtymää vastaa kaksi perättäistä symbolijonon alkioita. Tilasiirtymiä



Kuvio 1. Markovin ketju esitettynä tilakaaviona ja matriisina, mukailten Nierhaus (2009, s. 68)

ja tilojen lukumääriä laskettaessa symbolijonon ensimmäinen sekä viimeinen alkio jätetään huomioimatta (Gagniuc 2017, s. 28). Kuvion 1 tilasiirtymämatriisi saadaan muodostettua esimerkiksi symbolijonosta $AAAPASPPSSPPSAAA$, missä A tarkoittaa aurinkoista, P pilvistä ja S sadetta.

Markovin ketjuja, joissa tilasiirtymät riippuvat useammasta kuin yhdestä perättäisestä edellisestä tilasta, kutsutaan korkeamman kertaluvun Markovin ketjuiksi (Nierhaus 2009, s. 68). Mitä suuremman kertaluvun Markovin ketjua käytetään, sitä enemmän sillä tuotetut tulokset muistuttavat alkuperäistä aineistoa.

Markovin ketjuja on käytetty muun muassa eläinpopulaatioiden kasvun mallintamiseen (Moran 1953; Meyn ja Tweedie 1993, s. 5) ja Yleisradion Botsikot-verkkopalvelussa (Ojalehto 2013) tuottamaan fiktiivisiä uutisotsikoita. Markovin ketjut ovat myös perusta Markovin ketju Monte Carlo -algoritmeille.

3.2 Markovin piilomalli

Markovin piilomalleilla mallinnetaan piilossa olevien tilojen ja havaittujen tapahtumien sarjan suhdetta (Merwe ja Schulze 2011). Markovin piilomallissa taustalla olevan Markovin ketjun tilat ja tilasiirtymät ovat niin sanotusti piilossa eli ne eivät ole havaittavissa (Nierhaus 2009). Lopputulemat sen sijaan ovat havaittavissa. Lopputulemien perusteella pyritään tekemään päätelmiä piilossa olevista tiloista. Esimerkiksi kuviossa 2 säätilat (rainy, sunny) ovat piilossa olevia tiloja ja lopputulemia askareet (walk, shop, clean). Taustalla olevan Markovin ketjun alkutila määräytyy siten, että jokaisella tilalla on oma todennäköisyytensä tulla valituksi alkutilaksi.

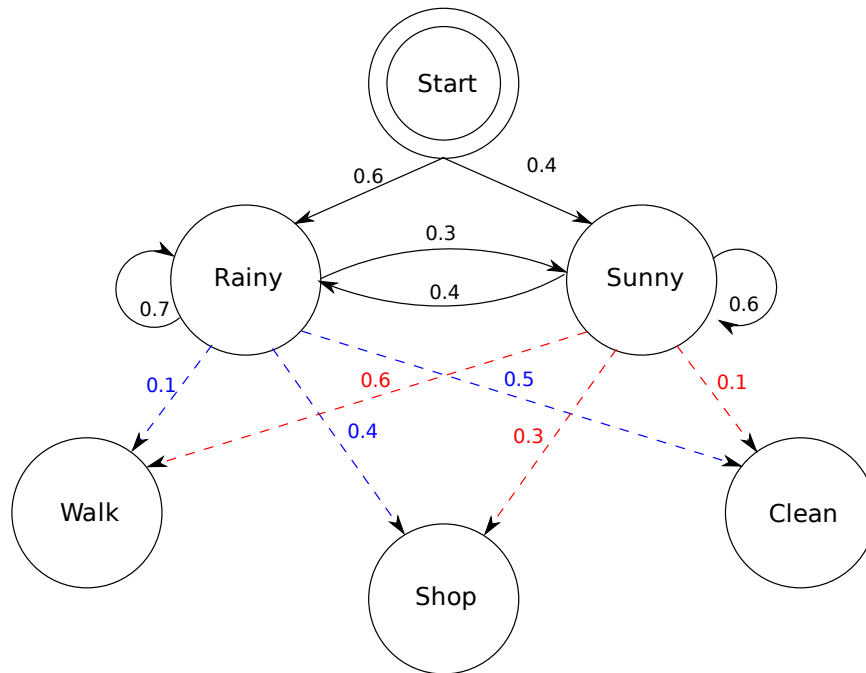
Markovin piilomalli määritellään usein viisikkona $(S, \Omega, P, \Phi, \pi)$, missä

- $S = \{s_1, s_2, \dots, s_N\}$ on äärellinen N -alkioinen piilossa olevien tilojen joukko;
- $\Omega = \{o_1, o_2, \dots, o_M\}$ on äärellinen M -alkioinen havaittavien lopputulemien joukko;
- $P = \{p_{ij}\}$ on tilasiirtymien todennäköisyyksien joukko, missä p_{ij} tarkoittaa siirtymätodennäköisyyttä tilasta s_i tilaan s_j ;
- $\Phi = \{\phi_i(o_k)\}$ on lopputulemien todennäköisyyksien joukko, missä $\phi_i(o_k)$ on todennäköisyys sille, että lopputulema on o_k järjestelmän ollessa tilassa s_i ;
- $\pi = \{\pi_i\}$ on alkutilojen todennäköisyyksien joukko, missä π_i on todennäköisyys sille, että tila s_i tulee valituksi alkutilaksi. (Ibe 2013, s. 419)

Kolmikkoa (P, Φ, π) kutsutaan Markovin piilomallin parametreiksi ja sitä merkitään kirjaimella λ (Ibe 2013, s. 419).

Markovin ketjut ovat moniin sovelluksiin huono valinta, sillä niiden tehokas käyttö edellyttää täyttä tuntemusta mallinnettavasta järjestelmästä sekä sitä, että järjestelmän toimintaa voidaan ohjata hyvin määritellyllä tavalla (Ibe 2013, s. 417). Jos edellä mainitut vaatimukset eivät täyty, Markovin piilomallit ovat varteenotettava vaihtoehto (Ibe 2013, s. 417). Markovin piilomalleille on käyttöä monilla sovellusalueilla, mutta erityisesti puheentunnistuksessa ja sen osa-alueilla (Jurafsky ja Martin 2008).

Markovin piilomallien yhteydessä käytetään yleensä kolmea algoritmia, jotka ovat forward-algoritmi, Viterbin algoritmi ja Baum-Welchin algoritmi (Nierhaus 2009). Näistä kukin vas-



Kuvio 2. Markovin piilomalli (Terencehonles 2009)

taa tiettyyn kysymykseen Markovin piilomallista. Forward-algoritmi laskee todennäköisyyden havaintosarjalle, kun Markovin piilomallin parametrit ovat tiedossa (Nierhaus 2009). Viterbin algoritmi laskee todennäköisimmän piilossa olevien tilojen sarjan, kun havaittujen tapahtumien sarja on tiedossa (Nierhaus 2009). Baum-Welchin algoritmia käytetään, kun annetun havaintosarjan perusteella halutaan löytää todennäköisimmät parametrit Markovin piilomallille (Nierhaus 2009). Baum-Welchin algoritmista käytetään myös nimitystä forward-backward-algoritmi (Ibe 2013, s.436).

3.3 Markovin ketjut algoritmisessa säveltämisessä

Algoritmisen säveltämisen kontekstissa Markovin ketjujen tilat voivat olla esimerkiksi nuotteja, sointuja tai lyhyitä fraaseja. Tilojen ollessa yksinkertaisia nuotteja yleinen käytäntö on ollut pitää niitä joko pelkkinä sävelkorkeuksina, nuottien kestoina tai niiden ja soittotavan, kuten voimakkuuden, yhdistelminä. Musikaalisessa kontekstissa voidaan tilojen määrä olettaa äärelliseksi. (Ames 1989)

Markovin ketjut voi muodostaa käsin tai ne voi johtaa aineistosta eli musikaalisessa konteks-

tissa musikaalista lähdemateriaalia analysoimalla (Ames 1989). Jälkimmäisellä lähestymistavalla pyritään yleensä jäljittelemään lähdemateriaalin musiikillista tyyliä. Lähdemateriaali voi olla esimerkiksi tietokoneella helposti luettavassa muodossa, kuten MIDI-tiedostoina. Jos musikaalisessa aineistossa on useita eri teoksia, tulee varmistua siitä, että ne ovat yhdenmukaisia, eli esimerkiksi samassa sävellajissa toistensa kanssa, sillä muuten analyysin tulokset eivät välttämättä ole tarkoituksenmukaisia (Ames 1989).

Tavalliset Markovin ketjut ovat helposti ja tehokkaasti toteutettavissa ohjelmallisesti (Ames 1989). Jo Hiller ja Isaacson (1959, s. 151) tekivät huomion, jossa heidän käyttämänsä tietokoneen nopeutta niitä käsitellessä rajoitti ainoastaan se, miten nopeasti se kykeni tulostamaan. Tehokkuus mahdollistaa myös reaaliaikaisten sovellusten toteuttamisen, kuten Continuatorin (Francois Pachet 2010) tapauksessa.

Markovin ketjuilla on pitkä historia algoritmisessa säveltämisessä, sillä niitä käytettiin jo johdannossa mainitun *Illiad* -suiteen osassa *Experiment Four* vuodelta 1956 (Nierhaus 2009, s. 72). Hiller ja Isaacson (1959, s. 141-142) muodostivat tilasiirtymämatriisin asettamalla kullekin intervallille todennäköisyyden sen konsonanssin eli sointusävyisyyden sekä suuruuden mukaan painottaen pienempiä intervaleja. Intervalli tarkoittaa kahden sävelen korkeuseroa. Konsonanssi on kahden yhtäaikaan soivan sävelen miellyttävyys. Experiment Fourissa käytettiin nollannen, ensimmäisen ja toisen asteen Markovin ketjuja sekä i-asteen Markovin ketjuja, joissa jokainen ensimmäistä tilaa (nuottia) seuraava tilasiirtymä riippui ensimmäisestä tilasta (Hiller ja Isaacson 1959, s. 146-147).

Pachet'n (2010) kehittämä *Continuator* on Markovin ketjuihin perustuva interaktiivinen järjestelmä, joka jatkaa käyttäjänsä soittoa siitä opittuaan. Continuator eroaa edeltäjistään interaktiivisen improvisoinnin saralla siten, että se ei tarvitse enempää informaatiota kuin pelkän käyttäjän reaaliaikaisen syötteen. Järjestelmä pilkkoo käyttäjän soitosta tulevan nuottivirran tietyin aikaväleihin, minkä jälkeen muodostetaan pätkien perusteella puumaisia tietorakenteita. Pachet'n (2010) mukaan tämä oppiva malli voidaan nähdä tehokkaana toteutuksena täydellisestä vaihtelevan kertaluvun (engl. complete variable-order) Markovin ketjusta.

Jotta Continuator mukautuisi käyttäjän soittoon tarpeeksi nopeasti, järjestelmässä käytetään pelkän todennäköisyyden lisäksi sovituskäytännön (engl. fitness function), joka painot-

taa käyttäjän viimeisintä syötettä (Francois Pachet 2010). Tämä mahdollistaa muun muassa nopeamman sointukulujen muutoksiin mukautumisen. Francois Pachet (2010) väittää, että järjestelmän tuotosta ei voi erottaa käyttäjän soitannasta etenkin nopeatempoisissa sooloissa. Menetelmää, jolla tähän lopputulokseen päädyttiin ei kuitenkaan artikkelissa kuvattu.

Konsensus vaikuttaa olevan siitä, että Markovin ketjut eivät sovellu kokonaisten sävellysten tuottamiseen. Tätä tukee muun muassa François Pachet (2012) toteamalla, että Markovin ketjuilta puuttuu pidemmän aikavälin ”muisti”. Nierhausin (2009) mukaan Markovin ketjut ovat parhaimmillaan sovellettuna yksiulotteisten symbolijonojen kanssa, mutta musiikki on luonteeltaan enemmän kerrostunutta ja kerrostuneisuuden huomioiminen saattaa laajentaa tilajoukkoa huomattavasti.

Korkeamman kertaluokan Markovin ketjut eivät näyttäyty ratkaisuna matalampien kertaluokkien huomattavalle musiikillisen rakenteen puutteelle sekä selkeälle satunnaisuudelle. Baffioni, Guerra ja Tedeschini-Lalli (1981) kritisoivat korkeamman kertaluokan Markovin ketjuja siitä, että kertaluokkaa kasvattaessa tilasiirtymämatriisin koko kasvaa ja musiikillinen anti alkaa muistuttaa enemmän läheaineistoa, minkä seurauksena sävellykset eivät ole niin omintakeisia. Myös Nierhaus (2009, s. 81) nostaa esiin samat ongelmat.

3.4 Markovin piilomallit algoritmisessa säveltämisessä

Musikaalisessa kontekstissa Markovin piilomallin lopputulemat ovat joitakin musiikin elementtejä, yleensä nuotteja, joilla on kesto ja sävelkorkeus (Nierhaus 2009). Markovin piilomallien lopputulemien jakaumat voivat olla diskreettejä tai jatkuvia, mutta musikaalisessa kontekstissa jatkuvia jakaumia on vähemmälti (Nierhaus 2009). Tämä johtunee siitä, että esimerkiksi sävelkorkeudet ja nuottien kestot pidetään usein diskreetteinä.

Markovin piilomalleja sekä Markovin ketjuja hyödyntää Merwen ja Schulzen (2011) kehittämä SuperWillow on järjestelmä, joka kykenee tuottamaan kaksiaänisiä sävellyksiä, joissa toinen ääni soittaa melodia ja toinen säestää. Järjestelmä pyrkii tuottamaan annettua aineistoa jäljitteleviä teoksia. SuperWillowssa sointujen kestot, sointukulut ja rytmit esitetään ensimmäisen, korkeamman tai sekalaisen kertaluokan (mixed-order) Markovin ketjuilla, joita sävellyksen tuottamisvaiheessa laajennetaan lisäämällä tiloja ja tilasiirtymiä siten, että tulok-

senä oleva kulku täyttää aina yhden tahdin. Melodian kulkua taas kuvataan Markovin piilomallilla, jonka taustalla on ensimmäisen, korkeamman tai sekalaisen kertaluokan Markovin ketju. (Merwe ja Schulze 2011)

Merwe ja Schulze (2011) järjestivät verkkokyselyn, jossa vastaajien oli arvioitava kolmea sävellystä, joista yksi oli ihmisen tuotos ja loput SuperWillow'n. Toisessa koneen tuottamassa sävellyksessä käytettiin korkeamman kertaluokan ja toisessa sekalaisen kertaluokan Markovin ketjuja. Vastaajien tuli tunnistaa sävellyksistä ihmisen tekemä sekä asettaa ne lempijärjestykseensä. Kyselyyn vastasi 263 henkilöä. Heistä 62 % tunnisti oikein ihmisen sävellyksen ja 59 % piti ainakin yhtä tietokoneen sävellystä ihmisen tuotosta parempana. Kyselyyn vastanneista 72 % joko ei kyennyt tunnistamaan ihmisen sävellystä tai piti jompaa kumpaa tietokoneen sävellyksistä ihmistä parempana. Merkittäviä eroja korkeamman ja sekalaisen kertaluokan Markovin ketjuille ei tuloksissa muodostunut. (Merwe ja Schulze 2011)

SuperWillow'n musiikin analysoinnin osuus on rajoittunut vain musiikkiin, jossa on kaksi instrumenttia ja ei tahtilajin muutoksia. Analysoinnin aikana tauot ja järjestelmälle tuntemattomat soinnut jätetään huomioimatta, mikä johtaa siihen, että järjestelmän tuottamassa musiikissa ei esiinny lainkaan taukoja. Kyselytutkimuksessa käytetyille ihmistekoisille sävellyksille asetettiin samat rajoitukset kuin järjestelmällä ja lisäksi vain aika-arvot kokonuo-
tista kuudestoistaosaan olivat sallittuja. Trioleja eikä yhdyssäveliä ei saanut käyttää. (Merwe ja Schulze 2011)

Jos ihmissäveltäjät olisivat saaneet löyhemmät rajoitukset, olisi lopputulos voinut olla hie-
man erilainen. Sävellykset olivat myös verrattain lyhyitä, vain kahdeksan tahtia, minkä Merwe ja Schulze (2011) myöntävät tietokoneen eduksi.

Yanchenko ja Mukherjee (2018) tutkivat Markovin piilomallien hyödyntämistä romantiikan aikakauden taidemusiikin jäljittelemisessä. Romantiikan ajan katsotaan musiikissa ajoittuvan 1700-luvun lopusta 1900-luvun alkuun (Yanchenko ja Mukherjee 2018). Yanchenko ja Mukherjee (2018) valitsivat kymmenen romantiikan aikaista pianolle sävellettyä tai sovitettua teosta, joista Markovin piilomallit luotiin. Markovin piilomalleista käytettiin tutkimuksessa useita eri variaatioita, kuten kerroksellista Markovin piilomallia (engl. layered Hidden Markov model) sekä korkeamman kertaluokan Markovin piilomalleja (engl. higher-order

hidden Markov model).

Kullakin luodulla mallilla tuotettiin tuhat sävellystä, jotka arvioitiin kvantitatiivisilla mittareilla, kuten esimerkiksi dissonanssi eli riitasointuisuus sekä alkuperäisen ja mallin luoman sävellyksen yhteisinformaation määrä. Näiden kvantitatiivisten mittareiden avulla valittiin kolme parasta sävellystä, joiden lähdeaineistona olivat Beethovenin Oodi ilolle (engl. Ode to Joy), Mendelssohnin Enkellaulu kajahtaa (engl. Hark! The Herald Angels Sing) ja Chopinin Pianosonaatti nro 2, osa 3 (surumarssi). Sävellyksistä ensimmäinen generoitiin ensimmäisen kertaluokan Markovin piilomallilla ja kaksi jälkimmäistä kerrostuneilla Markovin piilomalleilla. (Yanchenko ja Mukherjee 2018)

Parhaat malleilla tuotetut sävellykset soitettiin kuulijoille, joiden tehtävänä oli asettaa sävellykset lempijärjestykseensä ja vastata avoimiin kysymyksiin muun muassa siitä, voisiko sävellys olla ihmisen tekemä. Kuulijat eroteltiin niihin, jotka olivat musiikkiyhtyeen jäseniä, ja niihin, jotka eivät olleet, kahdeksan kutakin. Kuulijoiden mielestä yksikään generoiduista sävellyksistä ei muistuttanut romantiikan ajan musiikkia vaan ennemminkin nykyaikaista. Kuusi kuulijaa tunnisti yhden sävellyksen lähdeaineistona käytetyn Enkellaulu kajahtaa teosta, mikä on mahdollisesti merkki kerroksellisen Markovin piilomallin ylisovittamisesta. (Yanchenko ja Mukherjee 2018)

Kuulijoiden ja kvantitatiivisten mittareidenkin mukaan generoiduista sävellyksistä puuttui laajempi musikaalinen rakenne ja kehityskulku (Yanchenko ja Mukherjee 2018). Markovin piilomallit kärsivät siis samasta ongelmasta kuin Markovin ketjut. Yanchenko ja Mukherjee (2018) havaitsivat, että lähdeaineiston harmonisesti yksinkertaisista kappaleista tuotetut sävellykset sisälsivät enemmän konsonanssia ja kuulijat kokivat ne miellyttävämmiksi.

4 Yhteenveto

Markovin ketjuja ja Markovin piilomalleja käytetään algoritmisessa säveltämisessä pääasiassa musiikkityylien jäljittelyyn. Tähän tarkoitukseen menetelmät näyttävät heikkoina vaihtoehtoina, sillä niillä generoidut sävellykset saattavat muistuttaa joko liikaa tai liian vähän lähdeaineistona käytettyjä teoksia. Kuitenkin Markovin ketjuilla tai Markovin piilomalleilla tuotetut sävellykset voivat kuulostaa miellyttäviltä ja joissain tapauksissa ikään kuin ihmisen säveltämitä.

Molempien menetelmien avulla tuotetut sävellykset kärsivät usein laajemman musikaalisen rakenteen puutteesta. Musiikki ei siis kehity ajan mittaan vaan jäävät toistamaan samaa musiikillista ideaa. Tämän takia Markovin ketjut ja Markovin piilomallit sopivat paremmin lyhyempien sävellysten tuottamiseen.

Markovin ketjujen vahvuutena on niiden yksinkertainen toteuttaminen sekä tehokkuus, jonka ansiosta ne sopivat reaaliaikaisiin ja interaktiivisiin sovelluksiin. Markovin piilomallit sen sijaan ovat laskennallisesti vaativampia, joten ne eivät näihin tarkoituksiin välttämättä sovelly.

Tutkimisen arvoista on, voisiko Markovin ketjujen ja Markovin piilomallien avulla tuottaa musiikillisesti rakenteisempia sävellyksiä yhdistelemällä menetelmiä muihin. Markovin ketjuista ja Markovin piilomalleista on olemassa lukuisia muunnelmia, joista moni jäi tässä tutkielmassa käsittelemättä. Jos tavoitteena on parempi musiikkityylien jäljittely, lienee parasta turvautua joihinkin muihin algoritmisen säveltämisen menetelmiin.

Lähteet

Ames, Charles. 1989. "The Markov Process as a Compositional Model: A Survey and Tutorial". *Leonardo* 22 (2): 175–187. ISSN: 0024094X, 15309282. <http://www.jstor.org/stable/1575226>.

Baffioni, Claudio, Francesco Guerra ja Laura Tedeschini-Lalli. 1981. "Music and aleatory processes". Viitattu 21. huhtikuuta 2020. https://www.researchgate.net/profile/Laura_Tedeschini-Lalli/publication/266707447_Music_and_aleatory_processes/links/572dce5208aee022975a58d2/Music-and-aleatory-processes.pdf.

Biles, John A. 1994. "GenJam: A Genetic Algorithm for Generating Jazz Solos". *ICMC Proceedings*: 8. Viitattu 18. helmikuuta 2020. <https://genjamorg.files.wordpress.com/2019/07/bilesicmc94.pdf>.

Collins, Nick. 2008. "The Analysis of Generative Music Programs". *Organised Sound* 13 (3): 237–248. doi:10.1017/S1355771808000332.

Edwards, Michael. 2011. "Algorithmic Composition: Computational Thinking in Music". *Commun. ACM* 54, numero 7 (heinäkuu): 58–67. ISSN: 0001-0782. doi:10.1145/1965724.1965742.

Gagniuc, Paul A. 2017. *Markov Chains : From Theory to Implementation and Experimentation*. Wiley. ISBN: 978-1-119-38755-8. <http://search.ebscohost.com.ezproxy.jyu.fi/login.aspx?direct=true&db=nlebk&AN=1548386&site=ehost-live>.

Gracyk, Theodore, ja Andrew Kania. 2011. *The Routledge companion to philosophy and music*. Routledge.

Hedges, Stephen A. 1978. "Dice Music in the Eighteenth Century". *Music & Letters* 59 (2): 180–187. ISSN: 00274224, 14774631, viitattu 5. helmikuuta 2020. <http://www.jstor.org/stable/734136>.

- Hendrikx, Mark, Sebastiaan Meijer, Joeri Van Der Velden ja Alexandru Iosup. 2013. “Procedural Content Generation for Games: A Survey”. *ACM Trans. Multimedia Comput. Commun. Appl.* 9, numero 1 (helmikuu). ISSN: 1551-6857. doi:10.1145/2422956.2422957.
- Hiller, Lejaren, ja Leonard Isaacson. 1959. *Experimental music; composition with an electronic computer*. New York, McGraw-Hill. Viitattu 8. huhtikuuta 2020. <https://archive.org/details/experimentalmusi00hill/>.
- Ibe, Oliver C. 2013. *Markov Processes for Stochastic Modeling*. Nide 2nd edition. Elsevier. ISBN: 978-0-12-407795-9. <http://search.ebscohost.com.ezproxy.jyu.fi/login.aspx?direct=true&db=nlebk&AN=516132&site=ehost-live>.
- Jurafsky, Daniel, ja James Martin. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Nide 2. Viitattu 1. huhtikuuta 2020. https://www.researchgate.net/publication/200111340_Speech_and_Language_Processing_An_Introduction_to_Natural_Language_Processing_Computational_Linguistics_and_Speech_Recognition.
- Kania, Andrew. 2017. “The Philosophy of Music”. Teoksessa *The Stanford Encyclopedia of Philosophy*, Fall 2017, toimittanut Edward N. Zalta. Metaphysics Research Lab, Stanford University. Viitattu 4. helmikuuta 2020. <https://plato.stanford.edu/archives/fall2017/entries/music/>.
- Kotimaisten kielten keskus. 2020. *Kielitoimiston sanakirja*. Verkkojulkaisu HTML. Päivitetty julkaisu. Päivitetty 6.6.2018. Viitattu 2. huhtikuuta. <https://www.kielitoimistonsanakirja.fi/s%C3%A4velt%C3%A4%C3%A4>.
- Liu, C., ja C. Ting. 2017. “Computational Intelligence in Music Composition: A Survey”. *IEEE Transactions on Emerging Topics in Computational Intelligence* 1, numero 1 (helmikuu): 2–15. ISSN: 2471-285X. doi:10.1109/TETCI.2016.2642200.
- Mazzola, Guerino, Joomi Park ja Florian Thalmann. 2011. “Creativity in Composition and Improvisation”. Teoksessa *Musical Creativity: Strategies and Tools in Composition and Improvisation*, 233–247. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-24517-6, viitattu 17. maaliskuuta 2020. doi:10.1007/978-3-642-24517-6_22.

- Merwe, A. Van Der, ja W. Schulze. 2011. "Music Generation with Markov Models". *IEEE MultiMedia* 18, numero 3 (maaliskuu): 78–85. ISSN: 1941-0166. doi:10.1109/MMUL.2010.44.
- Meyn, S. P., ja R. L. Tweedie. 1993. *Markov Chains and Stochastic Stability*. Springer-Verlag. Viitattu 23. maaliskuuta 2020. <http://probability.ca/MT/>.
- Moran, P. A. P. 1953. "The statistical analysis of the Canadian Lynx cycle." Publisher: CSIRO Publishing, *Australian Journal of Zoology* 1 (3): 291. doi:10.1071/zo9530291.
- Nierhaus, Gerhard. 2009. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer, elokuu. ISBN: 978-3-211-75540-2. <https://www.dawsonera.com:443/abstract/9783211755402>.
- Ojalehto, Jaakko. 2013. "Robotti aloitti otsikkotehtailun - poliisi vastasi kumiluodeilla" (joulukuu). Viitattu 11. maaliskuuta 2020. <https://yle.fi/aihe/artikkeli/2013/12/10/robotti-aloitti-otsikkotehtailun-poliisi-vastasi-kumiluodeilla>.
- Pachet, Francois. 2010. "The Continuator: Musical Interaction With Style". *Journal of New Music Research* 32:333–341. Viitattu 14. huhtikuuta 2020. doi:10.1076/jnmr.32.3.333.16861.
- Pachet, François. 2012. "Musical Virtuosity and Creativity". Teoksessa *Computers and Creativity*, toimittanut Jon McCormack ja Mark d'Inverno, 115–146. Springer Berlin Heidelberg. ISBN: 978-3-642-31727-9. doi:10.1007/978-3-642-31727-9_5.
- Simon, Ian, Dan Morris ja Sumit Basu. 2008. "MySong: automatic accompaniment generation for vocal melodies" [kielellä en]. Teoksessa *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, 725. Florence, Italy: ACM Press. ISBN: 978-1-60558-011-1, viitattu 18. helmikuuta 2020. doi:10.1145/1357054.1357169.
- Terencehonles. 2009. *Hidden Markov Model*, marraskuu. Viitattu 8. huhtikuuta 2020. <http://commons.wikimedia.org/wiki/File:HMMGraph.svg>.

Wooller, Rene, Andrew Brown, Eduardo Miranda ja Joachim Diederich. 2005. "A Framework for Comparison of Process in Algorithmic Music Systems". Teoksessa *Generative Arts Practice*, toimittanut P. Brown, E. Edmonds ja D. Burraston, 109–124. Australia: Creativity & Cognition Studio Press. <https://eprints.qut.edu.au/6544/>.

Yanchenko, Anna K., ja Sayan Mukherjee. 2018. "Classical Music Composition Using State Space Models". ArXiv: 1708.03822, *arXiv:1708.03822 [cs]* (syyskuu). Viitattu 8. huhtikuuta 2020. <http://arxiv.org/abs/1708.03822>.