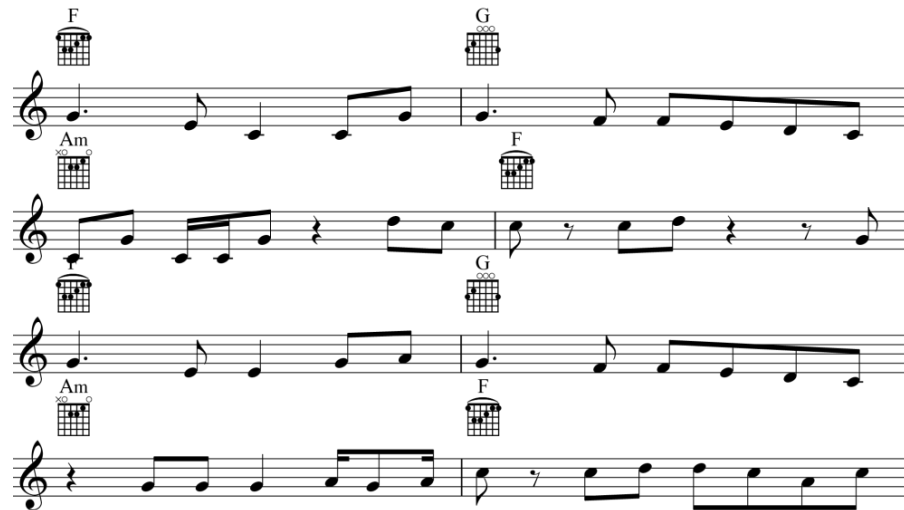


CHALMERS



Algoritmisk komposition av popmusik

– utformad med markovkedjor, bayesiskt nätverk samt strukturmodellering

Kandidatarbete inom Data- och Informationsteknik

Albin Lorentzson

Johan Grönvall

Karin Brötjefors

Per Olzon

Oscar Andersson

Viktor Wänerlöv

Chalmers tekniska högskola

Göteborgs universitet

Institutionen för Data- och Informationsteknik

Göteborg, Sverige, Juni 2015

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Algoritmisk komposition av popmusik

Albin Lorentzson
Johan Grönvall
Karin Brötjefors
Per Olzon
Oscar Andersson
Viktor Wänerlöv

© Albin Lorentzson, Juni 2015
© Johan Grönvall, Juni 2015
© Karin Brötjefors, Juni 2015
© Per Olzon, Juni 2015
© Oscar Andersson, Juni 2015
© Viktor Wänerlöv, Juni 2015

Examiner: Arne Linde

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Sammanfattning

Denna rapport föreslår och förklarar utvecklingen av en metod för att komponera modern popmusik algoritmiskt. Metoden använder maskininlärning och en strategi, som börjar med generering av struktur och avslutar med detaljer, för att generera ackordföljder och en passande melodi. I ett försök att fånga strukturen och självlikformigheten som existerar i popmusik genererar metoden en struktur i form av olika repetitioner. Ett test för att se om lyssnare kan urskilja genererade låtar från låtar komponerade av en människa visade att lyssnare hade svårighet att höra någon skillnad. Emellertid saknade låtarna fortfarande några viktiga egenskaper som resulterade i att en del av lyssnarna kunde se skillnaden.

Abstract

This study proposes and explains the development of a method for algorithmically composing modern pop music. The method uses machine learning in a top-down strategy to generate a chord progression and a fitting melody. In an attempt to capture the structure and self similarity present in pop music the method generates a structure in the form of partial repetitions. A test to see if a listener can tell a generated song from a human composed song showed that most listeners had trouble telling the difference. However, the songs still lacked some important properties resulting in some listeners being able to tell the difference.

Ordlista

Dynamiska bayesiska nätverk	– En grafisk modell som relaterar stokastiska variabler till varandra över intilliggande tidssteg.
Indikatorfunktion	– En funktion $f_A(x)$ som antar värdet 1 om $x \in A$ och 0 annars.
Lyssnarundersökning	– Ett test där ett antal personer får lyssna på musik och svara på frågor.
Markovkedja	– En diskret stokastisk variabel med Markovegenskapen.
Motiv	– En samling noter utsträckta över en takt.
Motivförändring	– När ett motiv liknar ett annat motiv till en viss utsträckning i ett musikstycke.
Oktav	– Ett intervall mellan två toner där frekvenserna har förhållandet 2:1.
Ram	– En halvtakt innehållande ett ackord och en samling noter.
Sektion	– En följd av fyra motiv.
Självlikformighet	– När ett musikstycke innehåller olika delar som liknar varandra.
Struktur	– Karakteristiken i en låt.
Sångbar	– En låt är sångbar om den består av en följd av noter som är lätt att sjunga.

Innehåll

1	Inledning	1
1.1	Bakgrund till algoritmisk komposition	1
1.1.1	Algoritmisk komposition idag	2
1.2	Syfte	2
1.3	Avgränsningar	2
2	Teori	4
2.1	Musikteori	4
2.1.1	Toner, skalor, ackord & rytm	4
2.1.2	Att komponera musik: melodier och ackordföljder	5
2.1.3	Att komponera musik: partier	7
2.2	Stokastiska modeller	7
2.2.1	Markovkedjor	7
2.2.2	Riktad acyklisk graf	9
2.2.3	Dynamiska bayesiska nätverk	10
2.3	Regelbaserade metoder	11
3	Metod	12
3.1	Databearbetning	12
3.1.1	Databasen: hooktheory	13
3.1.2	MIDI-format	13
3.1.3	Melodi- och ackordanalys	14
3.1.4	Strukturanalys	15
3.1.5	Rytmanalys	15
3.2	Modell och träning	16
3.2.1	Modell för struktur	16
3.2.2	Modell för ackordföljder	17
3.2.3	Modell för notföljder	18
3.3	Musikgenerering	19
3.4	Utvärdering av programkomponerad musik	20
3.4.1	Statistisk analys av popmusik	20
3.4.2	Lyssnarundersökning	21
4	Utvecklingsprocess	23
4.1	Generering av melodi med enkel markovkedja	23
4.2	Generering av melodi med markovkedja av högre ordning	24
4.3	Generering av melodi och ackord med dynamisk bayesiskt nätverk	24
5	Resultat	26
5.1	Genererad musik: exempel på olika musikstycken som programmet genererat	26
5.2	Statistisk jämförelse mellan databasen och den genererade musiken	27
5.3	Lyssnarundersökning: förhållandet mellan människo- och datorgenererad musik	29
6	Diskussion	32
6.1	Statistiska likheter och skillnader mellan genererade musikstycken och popmusik	32
6.2	Lyssnarundersökning: förhållandet mellan människo- och datorgenererad musik	33
6.3	Likheter och skillnader med tidigare studier	33
6.4	Övriga brister som programmet uppvisade och lösningar till dessa	35
7	Slutsatser	36

1 Inledning

Musik har varit och är en betydande del av det mänskliga samhället. Den har funnits med i alla kulturer och har förmågan att väcka känslor och förmedla budskap. Musikaliska verk påverkar olika individer på olika sätt och vissa skapar till och med så starka känslor att verken vill höras igen. För att kunna bevara, eller snarare återskapa, dessa känslor har musiken arkiverats både som notblad samt som inspelningar med till exempel en dator. Med tiden har datorn gått från att endast kunna spara och spela upp redan skapade verk till att även skapa dem.

Utvecklingen av datorer har fått människor att fundera över om en dator kan prestera lika bra eller bättre än en människa inom olika ämnen. Några ämnen som datorer har varit framgångsrika i är schack och frågesport [1]. Denna rapport inriktar sig mot användandet av datorer inom musik med fokus på musikkomposition. Det har skapats flertalet system som komponerar låtar i olika genrer, till exempel Illiac I där algoritmer användes för att hitta mönster inom färdigkomponerad musik och applicerade detta på nya verk[2].

Poängen med att komponera musik algoritmiskt är inte självklar, men oftast är det intressant att undersöka hur nära ett datorkomponerat musikstycke kan komma ett människokomponerat. En annan anledning är att det är ett sätt att förstå musiken bättre och på så sätt själv bli en bättre kompositör.

De flesta tidigare studier inom algoritmisk komposition behandlar genrer som jazz [3] och klassisk musik [4], vilket gör att denna rapport skiljer sig från dessa eftersom den endast behandlar popmusik. Denna rapport beskriver också ett program som genererar en övergripande struktur först för att sedan generera detaljer som ackord och melodi. De flesta andra studier brukar göra det i motsatt ordning [5].

Verktyg som användes för att skapa detta program var musikteori, stokastiska modeller och regelbaserade metoder. Musikteorin användes för att hitta mönster och förstå musiken. De stokastiska modeller som användes bestod främst av markovkedjor vars syfte var att generera nya musikföreteelser. Regelbaserade metoder nyttjades för att få fram rytmer. Utförandet bestod av en databearbetningsdel, en träningsdel och en genereringsdel där samtliga delar var uppdelade i tre kategorier: melodi, ackord och struktur. I databearbetningen analyserades en databas med en stor mängd låtar. Träningsdelen bestod av att använda datan från databearbetningen för att skapa en mall till genereringen. Under genereringen används sedan mallen för att skapa ett nytt musikaliskt verk. Slutligen testas ett antal genererade musikverk med en lyssnarundersökning.

För att kunna förstå rapporten i sin helhet förväntas läsaren besitta förkunskaper inom mängdlära och matematisk statistik.

1.1 Bakgrund till algoritmisk komposition

Grunden till algoritmisk komposition ligger i kopplingen mellan musik och matematik, ett samband som först dokumenterades av Pythagoras [6]. Att musik blev matematiskt modellerad gjorde det lättare att se dess mönster och det ledde till ett av de enklaste sätten att algoritmiskt komponera musik, med den kanoniska musiken. Denna kom på 1400-talet och bestod av att en kompositör komponerade ett kort musikstycke som flera personer sedan sjöng samtidigt, fast de började med en viss tidsförskjutning [7]. På 1700-talet tog algoritmisk komposition ett stort steg framåt när ett av de mest använda exemplen av algoritmiskt skapad musik, Musikalisches Würfelspiel, komponerades av Mozart (1756-1791). Detta verk skapade han genom att bestämma ordningen för diskreta musikutdrag genom att kasta två 6-sidiga tärningar [7]. Under 1900-talet ökade möjligheterna för algoritmisk komposition drastiskt tack vare datorns utveckling. Istället för att slå två 6-sidiga tärningar kunde nu datorn istället slumpa mellan många fler alternativ och ta hänsyn till många fler variabler än bara ett kort musikstycke, som i den kanoniska musiken. Illiac I var den första datorn som komponerade egen musik. Det första stycket döptes till Illiac suite (1957) som var skapat av flertalet olika algoritmer där syftet var att testa dessa algoritmer för att se om man kunde skapa ett tillfredställande musikverk [2].

1.1.1 Algoritmisk komposition idag

Inom algoritmisk komposition används idag en stor blandning metoder. En metod är att använda sig av grammatiken som finns inom musik. Detta görs genom att sätta upp regler för grammatiken, vilka sedan avgör hur verket kommer att genereras. Reglerna appliceras oftast i olika lager där lagren kan bestå av hur en musikgenre generellt låter till hur de olika individuella noterna ska arrangeras [5]. Problemet med en grammatisk ansats är att definiera reglerna som behövs för att skapa en kreativ komposition. Reglerna kan antingen tas fram för hand eller genereras automatiskt. R. M. Keller och D. R. Morrison[8] skapade 2007 en musikkompositör som skrev jazzsolon utifrån regelbaserad grammatik som han hittat för jazz.

Markovkedjor, något som testades redan 1975 [9], har använts på många olika sätt. Ett sätt är till maskininlärning för att skriva grammatik. Inom maskininlärning används redan komponerade verk som modellen sedan tränas på. Resultatet som erhålls är därmed lik musiken som analyserats. Ett annat sätt är att göra som Cruz-Alcázar och E. Vidal-Ruiz[10] gjorde när han analyserade skrivna verk för att skriva regler till grammatiken, vilka har använts till att komponera musikstycken [11]. Markovkedjor användes framförallt i början av utvecklingen av algoritmisk komposition, men frångicks senare då bättre metoder hittades. De förekommer dock fortfarande i hybrida system som använder sig av en blandning metoder.

Vanligare metoder för algoritmisk komposition idag är artificiella neurala nätverk (ANN) och evolutionära metoder. ANN är inspirerad av hjärnans biologiska nätverk och används inom maskininlärning. I början användes ANN enbart för musikstrukturer [12], men har även använts för komposition. Hybridsystem, där ANN används tillsammans med andra metoder, är välanvänt. Verbeurg et al.[13] skrev 2004 ett program som analyserade strukturen på låten med ett ANN och sedan använde en markovkedja för att analysera följderna av noter. En annan vanlig metod är att kombinera ANN och evolutionära metoder.

Evolutionära metoder använder ett antal olika lösningsförslag som genomgår ett upprepat urval och reproduktion med variationer. Lösningsförslagen mäts sedan med en lämplighetsfunktion för att bestämma dess kvalitet. Ett sätt att implementera lämplighetsfunktionen är med hjälp av en viktad summa, vilken består av egenskaperna som kompositionen ska innehålla. Marques et al.[14] skapade 2000 ett program som komponerade polyfonisk musik med denna metod. Harris och J. Fitch[15] försökte istället göra ett program vars uppgift var att skriva jazzsolon utifrån en specifik ackordföljd.

Lösningar som är specialiserade på en specifik musikgenre förekommer också. Framförallt är det vanligt med metoder som är anpassade för jazzimprovisationer och klassisk musik. Denna rapport skiljer sig eftersom den behandlar en metod som är utformad för modern popmusik.

1.2 Syfte

Rapporten föreslår och förklarar utvecklingen av ett program för att algoritmiskt komponera popmusik. Programmet använder maskininlärning som baseras på ett dynamiskt bayesiskt nätverk. Inlärningen görs från en databas bestående av redan existerande poplåtar. Utifrån dessa låtar komponeras sedan nya musikstycken. Resultatet av metoden utvärderas med en hjälp av en lyssnarundersökning och en statistisk analys.

1.3 Avgränsningar

Rapporten behandlar ett program som enbart genererar musikstycken i fyrtakt som består av två, tre eller fyra sektioner, där varje sektion består av fyra takter.

Kompositionerna som skapas av programmet komponeras endast i två stämmor, vilka är melodi och ackord. Programmet behöver inte kunna generera dessa i realtid och varken tidskomplexiteten eller minneskomplexiteten för de algoritmer som används behöver optimeras, så länge som programmet kan exekveras.

Rapporten menar inte att fånga de kreativa ändringar som mänskliga musiker tillför musiken vid ett framträdande, utan endast att fånga själva kompositionen.

2 Teori

I denna del presenteras den teori som ligger till grund för de metoder som utnyttjas i utvecklingen av programmet. Först presenteras musikteori i avsnitt 2.1 för att ge grundläggande förståelse i musiks uppbyggnad. Vidare presenteras stokastiska modeller i avsnitt 2.2, vilka utgör grunden för de matematiska modellerna till musik. Till sist presenteras regelbaserade metoder i avsnitt 2.3 eftersom dessa används vid genereringen av rytm.

2.1 Musikteori

För att skapa ett program som kan skriva popmusik är det bra att veta vad som särskiljer popmusik från annan musik. Enligt Frith et al. [16], är ett av de utmärkande dragen för popmusik att den är tänkt att tilltala den breda massan, därav namnet ”populär” musik. Popmusiken ska också följa trender som redan existerar och inte utveckla något nytt inom musikskapandet [17]. I denna del av teorin presenteras några grundläggande delar av musikteorin bakom popmusik och vad programmet ska eftersträva för att låta som popmusik.

2.1.1 Toner, skalor, ackord & rytm

I sitt verk ”Matematik & musik” förklarar Ulin [18] hur musik är uppbyggd ur ett matematiskt och fysikaliskt perspektiv. Musik byggs upp av toner som uppstår då föremål vibrerar. Då vibrationen sker skapas en rörelse i luften omkring föremålet och denna rörelse fortplantas genom luften och in i våra öron. Toner har därmed en frekvens som överensstämmer med föremålets vibrationsfrekvens och det spann av frekvenser som människan kan höra ligger mellan 20 och 20000 Hertz.

Vidare beskriver Ulin [18] hur en skala är uppbyggd av ett antal toner. Den vanligaste skalan inom västvärldens musik är tolvtonsskalan där oktaven delas upp i tolv olika toner, se figur 1. Från denna tolvtonsskala kan sedan skalor med färre toner formars. Några vanliga skalor är durskalan och mollskalan, som båda är uppbyggda av sju toner var. Även dessa presenteras i figur 1. Skillnaden mellan de två skalorna är att i mollskalan har den tredje, sjätte och sjunde tonen sänkts. Oftast associeras durskalan till gladare musik medan mollskalan uppfattas som sorglig [18]. Vidare finns det många fler skalor där de olika tonerna i tolvtonsskalan varieras.

Tolvtonsskalan



Durskalan



Mollskalan



Figur 1: De vanligaste skalorna inom västvärldens musik. Högst upp demonstrerar tolvtonsskalan alla toner som används i västvärldens musik. I mitten visas durskalan, en skala som används vid skrivning av glada melodier. Längst ned presenteras mollskalan som oftast associeras till mörkare och sorgligare melodier. Skalan skiljer sig från durskalan på sättet att den tredje, sjätte och sjunde tonen är sänkta ett halvt tonsteg vilket skapar den mer sorgsna tonen.

Ulin [18] beskriver även hur ackord är uppbyggda. Då flera toner spelas samtidigt bildas ett så kallat ackord. Ackordet består alltid av en grundton och beroende på färgning av ackordet väljs sedan resterande toner. De vanligaste ackorden består av tre toner; den första, den tredje och den femte tonen i skalan. Dessa kallas även grundtonen, tersen och kvinten. Som ett exempel spelas ackordet C-dur med tonerna C, E och G.

Rytm skapas av skillnaden mellan toners längd och pauser mellan dessa och även detta förklarar Ulin [18]. För att organisera rytmer delas låtar upp i takter där ett visst antal taktslag ska rymmas. Ett taktslag kan jämföras med ett pulsslag i hjärtat. Om tre sådana ska vara en takt fås 3-takt. Popmusik skrivs oftast i 4-takt, det vill säga med fyra slag per takt. Tonerna kan då vara en hel takt långa eller kortare så att fler toner kan spelas i samma takt.

2.1.2 Att komponera musik: melodier och ackordföljder

Under föregående rubrik presenterades de byggstenar som musik består av. Dessa kan kombineras för att skapa hela kompositioner. För att illustrera detta används delar ur Katy Perrys, "Last Friday Night (T.G.I.F)", en poplåt skriven år 2010. I figur 2 visas bitar ur versen och refrängen i låten.

Last Friday Night vers

Last Friday Night refräng

(a) I versdelen går det att se en repetitiv melodislinga som är en takt lång, vilket visas med de gröna rutorna. Den mörkgröna rutan är i princip samma melodislinga fast med en liten variation. De orangea rutorna visar hur dessa takter sedan återupprepas en gång till på samma sätt. Detta är ett vanligt sätt att bygga upp verser inom popmusik.

(b) I refrängdelen finns en ensam takt som sedan följs av tre identiska takter. Den första takten, som är markerad med en blå ruta, innehåller sångens krok, där låtens titel sjungs under en melodislinga. Denna melodislinga återkommer sedan flera gånger i refrängen. Som i versen upprepas fyra takter om och om igen för att bilda hela refrängen.

Figur 2: Versen och refrängen i låten "Last Friday Night" som framförs av Katy Perry. Låten är ett bra exempel på typiska delar som finns inom popmusik och hur melodier är strukturerade.

Inom popmusik är melodin en viktig del eftersom det oftast är den som framhävs av resten av musiken [19]. Genom att placera flera toner, med olika rytmer, efter varandra bildas melodin. Oftast har melodin ett tema med ett antal toner som sedan upprepas med små variationer[19]. I exemplet med "Last Friday Night", i figur 2, går det att observera dessa upprepningar. De takter som är markerade med ljusgrönt är exakta upprepningar av varandra och i den sista takten sker upprepningen med en liten variation. Det går även i exemplet att se att det inte bara är takter som upprepas utan även större partier om 4 takter, något som även det är vanligt inom popmusiken. Det handlar om att hitta en bra nivå mellan repetition som knyter samman låten och variationer som gör låten intressant [20].

Ett annat viktigt element inom popmusik beskrivs i boken "You can write a song" [19] av Appleby och är så kallade "hooks", översatt till svenska som "krokar". En krok är en melodislinga som återkommer ofta i musiken. Det är vanligt att kompositörer börjar med kroken och sedan skriver resten av melodin kring denna. Oftast är texten som sjungs under kroken densamma som titeln och är även det som minns tydligast efter att ha lyssnat på låten. I exempelsången är låtens krok markerad med blått.

I exemplet ses också att ackorden upprepas i följder. Alla poplåtar innehåller ackordföljder och vissa används oftare än andra [20]. Vad som gör att vissa ackordföljder är mer använda än andra är oklart, mycket handlar om vana och vad som känns bra för lyssnaren. Ackordföljder är en viktig del inom musikskapandet eftersom den sätter stämningen för låten [20].

2.1.3 Att komponera musik: partier

En annan viktig del vid skapandet av popmusik är låtens olika delar eller partier. I poplåtar finns oftast tydliga delar som har olika uppgift i låten. Enligt Appleby [19] är de vanligaste delarna:

- Introt, vars uppgift är att inleda låten, fånga lyssnarens intresse och presentera en melodislinga som sedan återkommer genom låten.
- Versen, som ofta sätter känslan på låten. Oftast berättar texten någon typ av historia och introducerar då temat och vilka karaktärer eller liknande som är involverade. Versen ska även bygga upp till det huvudsakliga i låten som kommer i refrängen.
- Bryggan är en förlängning av versen och brukar oftast användas för att knyta samman versen och refrängen.
- Refrängen, den viktigaste delen i låten och som resten av delarna försöker framhäva. I refrängen brukar oftast låtens krok finnas. Budskapet i refrängen ska även vara lätt för lyssnaren att ta till sig.
- Sticket är den del som ofta skiljer sig från resten av låten. Genom att bryta den känslan som vers och refräng skapat kan lyssnarens intresse för låten upprätthållas. Sticket leder sedan in till versen eller refrängen.

Långt ifrån alla låtar innehåller alla dessa delar men som en grund då en poplåt skrivs kan det vara bra att få med dessa.

Detta är bara ett liten del i all den musikteori som finns kring popmusik men innefattar ändå de viktigare delarna man kan tänka sig att ett program som skapar popmusik borde klara av att producera [18][19][20].

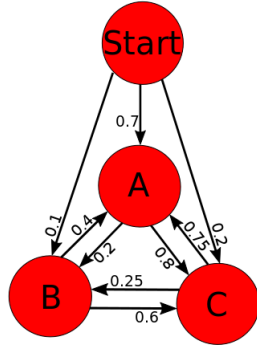
2.2 Stokastiska modeller

Datorprogrammet som denna rapport behandlar utnyttjar stokastiska metoder för att komponera musik. Dessa stokastiska metoder används för att modellera musiken och beskrivs i detta avsnitt.

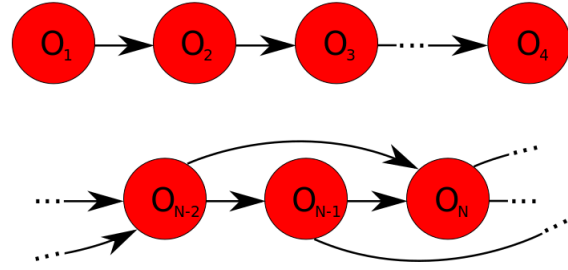
Först behandlas markovkedjor i avsnitt 2.2.1, vilka är stokastiska processer med glömska. Avsnittet 2.2.2 beskriver hur grafer fungerar, vilket är nödvändigt för att förstå dynamiska bayesiska nätverk (DBN). Till sist behandlas dynamiska bayesiska nätverk i avsnitt 2.2.3 som används när sannolikhetsberoenden mellan olika sorters objekt ska modelleras.

2.2.1 Markovkedjor

En markovkedja är ett sätt att modellera objekt som slumpmässigt förändrar sitt tillstånd mellan diskreta steg [21]. Objekten kan vara diverse saker, till exempel melodin i ett musikstycke. Melodin i ett musikstycke kan ses som ett objekt vars värde är en samling noter där varje övergång mellan noterna har olika sannolikhet. Figur 3a visar övergångarna mellan olika tillstånd för en sådan modell. Pilarna i figuren har nummer som representerar sannolikheten att förflyttas mellan tillstånden som pilarna kopplar ihop. I figur 3b illustreras beroendet mellan objektets värde i en viss tidpunkt med objektets värde i tidigare tidpunkter. För att fullständigt förstå en markovkedja måste stokastiska processer först förklaras.



(a) Figuren illustrerar hur de olika övergångarna som sker i en markovkedja kan se ut där startnoden innebär att inga tillstånd har antagits. Efter startnoden kommer objektet som observeras "hoppa" mellan olika tillstånd och till slut bildas en kedja över de olika övergångarna som skett. Det är dessa möjliga kedjor av händelser som kallas för en markovkedja. Bokstäverna i denna figur talar om vilket tillstånd som objektet befinner sig i och pilarnas värden anger sannolikheten för övergång mellan olika tillstånd.



(b) Den övre grafen i figuren illustrerar en första ordningens markovkedja, och den undre grafen illustrerar en andra ordningens markovkedja. Varje cirkel i figuren representerar en stokastisk variabel med ett utfallsrum vars värden är betingade av de cirklar som pekar på den observerade cirkeln. O_i är en stokastisk variabel för det i :te elementet i en kedja av händelser.

Figur 3: Två figurer som illustrerar olika egenskaper för markovkedjor. Den högra figuren visar beroendet mellan objektens tidigare tillstånd med det nuvarande tillståndet. Den vänstra figuren visar hur ett objekt förändrar sitt tillstånd med tiden.

Enligt Hsu [21] är en stokastisk process en samling slumpvariabler $\{X(t), t \in T\}$ definierade i ett sannolikhetsrum där en slumpvariabel indexeras t , där t varierar över en indexmängd T . En slumpvariabel är definierad i ett utfallsrum, vilket innebär att en stokastisk process är en funktion med två argument $\{X(t, \zeta), t \in T, \zeta \in S\}$. S är tillståndsrummet till de stokastiska variablerna.

Hsu[21] beskriver en markovkedja, med diskreta parametrar, som en stokastisk process med diskreta tillstånd som har Markovegenskaper. Detta innebär

$$P(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1) = P(X_{n+1} = j | X_n = i) \quad (1)$$

för en första ordningens markovkedja. Sannolikheten för att j ska ske beror alltså endast på vad som precis har skett. En markovkedja är en stokastisk process med glömska, den kommer inte ihåg vad som hände längre tillbaka i tiden. En markovkedja av ordning k blir

$$\begin{aligned} P(X_{n+1} = j | X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_1 = i_1) = \\ = P(X_{n+1} = j | X_n = i_n, X_{n-1} = i_{n-1}, \dots, X_{n-k+1} = i_{n-k+1}). \end{aligned} \quad (2)$$

Om processen är oberoende av värdet på n säger Hsu[21] att markovkedjan har stationära övergångar och processen kallas då homogen. Om markovkedjan är homogen och har ett ändligt stort tillståndsrum $E = \{0, 1, 2, \dots, m\}$, där beteckningen E brukar användas för diskreta tillståndsrum, kan vi definiera en övergångstensor av dimension ett, en övergångsmatrix, P givet $p_{ij} = P(X_{n+1} = j | X_n = i)$ som blir

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \dots & \dots & \dots & \dots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{pmatrix}. \quad (3)$$

Denna matris uppfyller villkoren

$$p_{ij} \geq 0 \quad \sum_{j=1}^m p_{ij} = 1 \quad i = 1, 2, \dots, m. \quad (4)$$

Om markovkedjan har en högre ordning k , där $k \geq 2$ kommer matrisen att bytas ut mot en tensor med dimension $k + 1$.

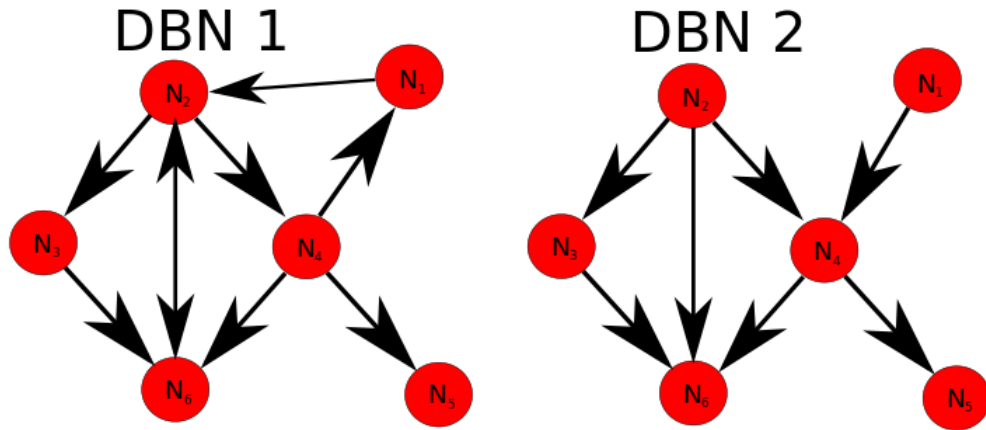
$$T_{i_n i_{n-1} \dots i_{n-k+1}} = P(X_n = i_n | X_{n-1} = i_{n-1}, \dots, X_{n-k+1} = i_{n-k+1})$$

I exemplet där melodin är en markovkedja kommer dess stokastiska variabler bestå av de noter som bygger upp melodin. En markovkedja kan alltså i exemplet ses som en samling notsekvenser av tillstånd där varje sekvens har olika sannolikhet att inträffa.

2.2.2 Riktad acyklisk graf

Enligt Leistikow [22] är en graf G en samling noder och kopplingar: $G = (Noder, Kopplingar)$. Noder är mängden av alla noder Nod_i och Kopplingar är mängden av alla nodpar (Nod_i, Nod_j) i grafen som har en koppling. En nod är riktad om $(Nod_i, Nod_j) \in Kopplingar$, men $(Nod_j, Nod_i) \notin Kopplingar$ och skrivs $Nod_i \rightarrow Nod_j$. I figur 4 illustreras två grafer med noderna $N_1, N_2, N_3, N_4, N_5, N_6$ samt kopplingarna $(N_2, N_3), (N_2, N_4), (N_2, N_6), (N_3, N_6), (N_4, N_5), (N_4, N_6)$ i respektive graf. Grafen till höger har också kopplingen (N_1, N_4) medan till vänster finns ytterligare kopplingar $(N_1, N_2), (N_4, N_1)$ samt (N_6, N_2) . Detta innebär att för grafen till höger i figuren gäller: $Noder = \{N_1, N_2, N_3, N_4, N_5, N_6\}$ och $Kopplingar = \{(N_1, N_4), (N_2, N_4), (N_2, N_6), (N_2, N_3), (N_3, N_6), (N_4, N_6), (N_4, N_5)\}$. Ett exempel på en koppling inte är riktad är (N_2, N_6) i vänstra grafen i figur 4.

De noder som pekar på en och samma nod kallas för föräldrar till noden de pekar på [22]. De noder som en enskild nod pekar på kallas tillsammans för barn till noden som pekar på dem. Föräldrarna till en nod, Nod_i , kallas $Par(Nod_i)$ och barnen till Nod_i kallas $Child(Nod_i)$. Detta skulle innebära att för nod N_3 till högra grafen i figur 4 gäller att: $Par(N_6) = \{N_2, N_4, N_3\}$, $Child(N_2) = \{N_3, N_4, N_6\}$.



Figur 4: Figuren illustrerar två olika grafer. Till höger en riktad acyklisk graf och till vänster en graf som varken är riktad eller acyklisk. Noderna N_1 , N_2 och N_4 i den vänstra grafen bildar en cykel vilket medför att denna graf inte är acyklisk. Noderna N_2 och N_6 i den vänstra grafen har en koppling som inte är riktad vilket medför att denna graf inte är riktad. Det är dessa två uppräknade egenskaper som skiljer de två graferna åt.

En riktad väg mellan Nod_i och Nod_j är en sekvens av olika noder $v_{1:n}$, där $v_1 = Nod_i$ och $v_n = Nod_j$. Visuellt kan en markör börja på Nod_i och följa pilarna mellan noderna tills markören placeras på Nod_j . Vägen som markören rör sig skulle då vara en riktad väg mellan Nod_i och Nod_j om den aldrig passerar en och samma nod mer än en gång. I högra grafen i figur 4 visas $N_1 \rightarrow N_4 \rightarrow N_5$, ett exempel på en riktad väg mellan N_1 och N_5 . Om det finns en riktad väg där $Nod_i = Nod_j$ kallas detta för en riktad cykel. En cykel innebär rent visuellt att pilarna följs från Nod_i tills samma nod erhålles. Ett exempel på en riktad cykel kan ses i figur 4 där grafen till vänster har den riktade cykeln $N_1 \rightarrow N_2 \rightarrow N_4 \rightarrow N_1$. En riktad acyklisk graf¹ är en graf där alla kopplingar till grafen är riktade och där riktade cykler inte förekommer.

2.2.3 Dynamiska bayesiska nätverk

Ett dynamiskt bayesiskt nätverk (DBN) är enligt Leistikow [22] en riktad acyklisk graf vars noder representerar stokastiska variabler och där sannolikhetsfördelningarna för dessa variabler är betingad av nodföräldrarnas stokastiska variabler. Relationen blir därmed

$$P(N_{n+1}|N_n, N_{n-1}, \dots, N_1) = P(N_{n+1}|Par(N_{n+1})). \quad (5)$$

Detta innebär att hela systemets sannolikhetsfördelning är

$$P(N_1, N_2, \dots, N_n) = \prod_{i=1}^n P(N_i|Par(N_i)) \quad (6)$$

Med dynamiska bayesiska nätverk kan en sekvens av objekt genereras genom att generera ett objekt i taget. När senare objekt genereras i en sekvens beror de på vilka värden som antagits av objekten som skapats

¹förkortat DAG-directed acyclic graph

tidigare i samma sekvens. Nätverket är alltså ett verktyg för modellering av vilka beroenden som finns mellan olika komponenter i ett större system. I denna studie består dessa objekt av noter till melodin samt ackord.

Rent praktiskt vad denna modelleringsmetod innebär vid generering är att startnodernas värden genereras först. Startnoder är noder utan några nodföräldrar. Efter att dessa noders värden skapas genereras värden för de noder som startnoderna pekar på. Vidare genereras värden till de noder vars nodföräldrars värden redan skapats tills samtliga noder har fått ett värde.

2.3 Regelbaserade metoder

Regler är en viktig del inom det vardagliga livet, vetenskap och musik. Människor har olika regler för hur man behandla varandra och vetenskapen har regler för hur världen fungerar. Den mest grundläggande formen på en regel är [23]

$$Regel : Förhandsvillkor \rightarrow Resultat$$

Inom musiken finns det regler för hur en viss typ av stil ska spelas, vilka toner som passar ihop med vilka ackord etc. Det finns många regler inom musiken, se avsnitt 2.1, där vissa av reglerna är strikta regler medan andra är informella [23]. De strikta reglerna finns nedskrivna, medan de informella är något som hörs men är svåra att nedteckna. En stor restriktion med regelbaserade metoder är att det kan vara mycket svårt att hitta de informella reglerna, vilket gör att de sällan blir perfekta.

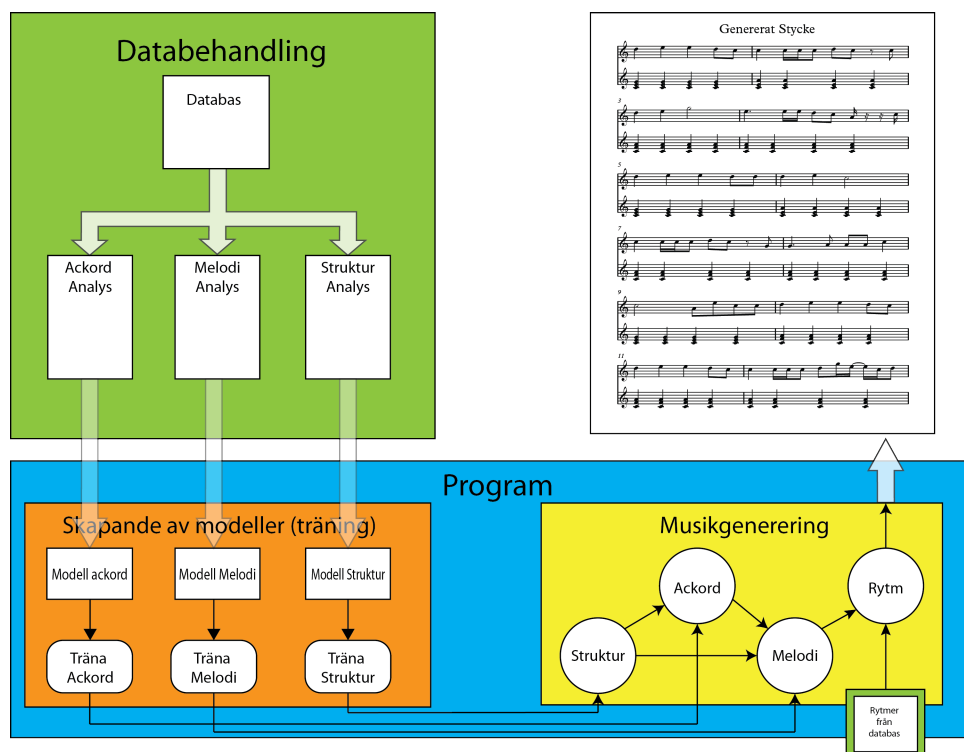
Det finns även regler där förhandsvillkoren är samma, men som har olika resultat. Det kan då finnas sannolikheter för att de olika resultaten inträffar. Sannolikheterna kan fås ur data från låtar, från musikteori för pop eller genom att själv välja parametrarna.

3 Metod

Som inledningen tidigare beskrev är det vanligt inom algoritmisk komposition att börja med att generera detaljer så som melodislingor och utifrån dem skapa en låt med ackord samt struktur. I detta avsnitt beskrivs en alternativ metod för att generera musik där en övergripande del av verket, musikstyckets struktur, först genereras för att sedan fyllas med detaljer så som ackord och melodislingor.

Tillvägagångssättet för hur programmet genererar musik är sammanfattat i figur 5. Först analyseras en mängd redan komponerad musik på tre olika sätt: vilka ackord som spelas, vilka tonföljder som används för att skapa en melodi samt vilka strukturer som tillämpas i styckena. Därefter tränas modellerna i programmet på datan så att de olika delarna kan skapas. Slutligen genereras musik utifrån modellerna i ordningen som visas i figuren.

I de kommande avsnitten kommer dessa delar presenteras noggrannare i den ordning figur 5 visar.



Figur 5: Sammanfattning för tillvägagångssättet att komponera musiken. Först analyseras popmusik från en databas och med denna data kan programmet sedan tränas och skapa modeller som därefter används vid genereringen av musiken.

3.1 Databearbetning

I detta avsnitt presenteras val och behandling av en databas med poplåtar. En databas var nödvändig eftersom programmet använder sig av maskininlärning. Det var många aspekter som behövde tas i beaktande vid valet av data. Den behövde vara av tillräckligt stor kvantitet, lätthanterlig och vara korrekt på ett sådant sätt att den speglar verk komponerade av människor. Vidare krävdes att programmets dataanalys är kompatibel med datan i databasen.

3.1.1 Databasen: hooktheory

Databasen som används av programmet kommer från Hooktheory, vilket är ett projekt för att undersöka musikteori samt att på ett visuellt och lättförståeligt sätt lära ut musikteori. Denna databas valdes eftersom den har vissa fördelar i jämförelse med andra studerade databaser innehållandes popmusik.

En av fördelarna är att låtarna i databasen är av formatet MIDI. MIDI-formatet och dess fördelar kommer förklaras noggrannare under nästa rubrik, avsnitt 3.1.2. Att använda sig av andra alternativa ljudformat, exempelvis mp3, skulle kräva djupa kunskaper i frekvensanalys eftersom samtliga stämmor ligger sammanmixade i ett enda spår. I MIDI-formatet ligger däremot stämmorna på separata spår, vilket gör det möjligt att direkt analysera dem var för sig. Eftersom rapporten beskriver ett program vars syfte är att komponera musik, inte analysera data, var en MIDI-databas att föredra.

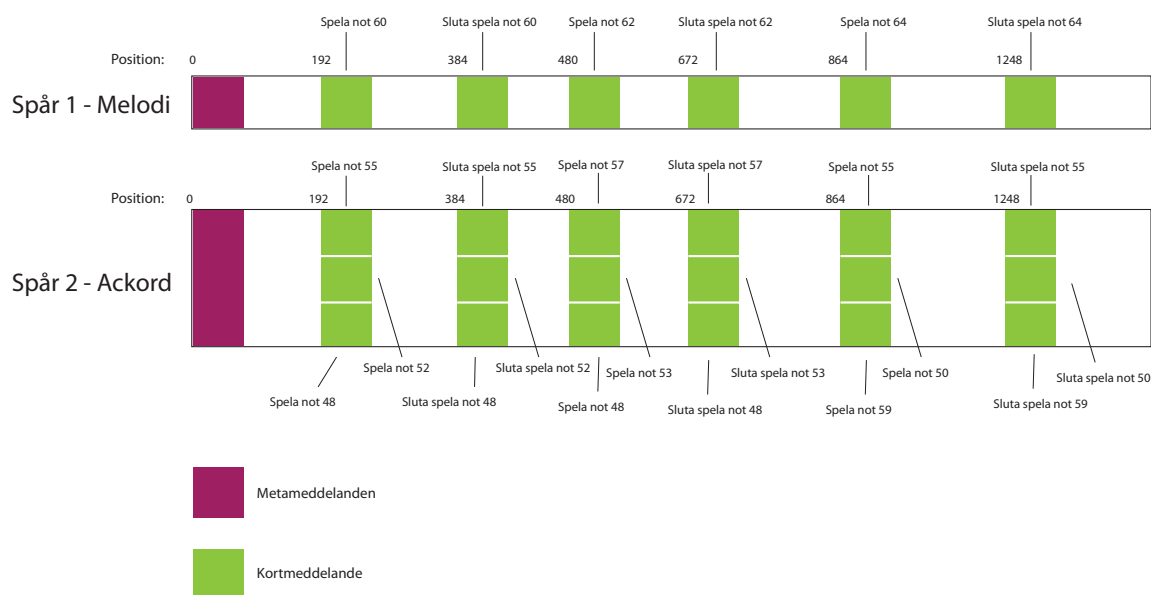
Ytterligare en fördel, som förklaras tydligare i kommande avsnitt, är att samtliga låtar kan hämtas i samma tonart. I andra databaser finns oftast låtarna i den tonart som originalet spelas i och nackdelen med detta är att förekomsten av olika toner förekommer i varierande grad i alla tonarter. Den gemensamma tonarten C valdes för samtliga låtar, vilket förenklade samtliga delar i programmet.

Det finns även nackdelar med databasen. Den var uppdelad i partier vilket medförde problem eftersom den saknar information om övergångar mellan olika partier i låtarna och därför kunde detta inte återskapas.

3.1.2 MIDI-format

Filerna i databasen är på formatet MIDI, vilket står för "Musical Instrument Digital Interface". MIDI är ett format som funnits i över 30 år. För att kunna hantera filerna var det nödvändigt att analysera formatets uppbyggnad. Därför presenteras nu hur datan förvaras i MIDI och hur den kan extraheras.

En översikt över hur MIDI-formatet ser ut illustreras i figur 6. MIDI-filers största beståndsdel är så kallade spår som innehåller de olika stämmorna i ett musikstycke. När en MIDI-fil spelas upp spelas samtliga spår på varandra och därmed spelas hela musikstycket. Ett spår består i sin tur av olika händelser. En händelse har två parametrar: en position och ett meddelande. Positionen markerar vart i spåret som händelsen inträffar och alla händelser sker i kronologisk ordning. Meddelandet kan vara av två typer, antingen är det ett meta-meddelande eller ett snabbmeddelande. Metameddelanden innehåller information om hela musikstycket, till exempel tempo eller vilka instrument som spelas. Snabbmeddelanden ger istället information som beskriver musiken, vilken ton som spelas och om tonen ska påbörjas eller avslutas. Om en ton påbörjas kommer den att fortsätta spelas tills det kommer ett nytt snabbmeddelande om att tonen ska avslutas. Flera toner kan spelas samtidigt om positionerna för händelserna överlappar. I snabbmeddelande representeras varje ton av en siffra, till exempel motsvaras C4 av siffran 60. För att generera MIDI-musik skapas alltså händelser som innehåller olika kommandon.



Figur 6: Midiformat går att visualisera som en tidslinje. På denna linje finner man olika händelser. Dessa kan antingen innehålla metameddelande eller snabbmeddelanden. Metameddelanden kan innehålla information om hela musikstycket, vilket tempo låten har eller vilka instrument som spelas. Snabbmeddelanden innehåller information om en not ska spelas eller sluta spelas och avståndet mellan dessa bestämmer hur lång noten är. I figuren visas hur de olika spåren kan se ut för melodi och ackord.

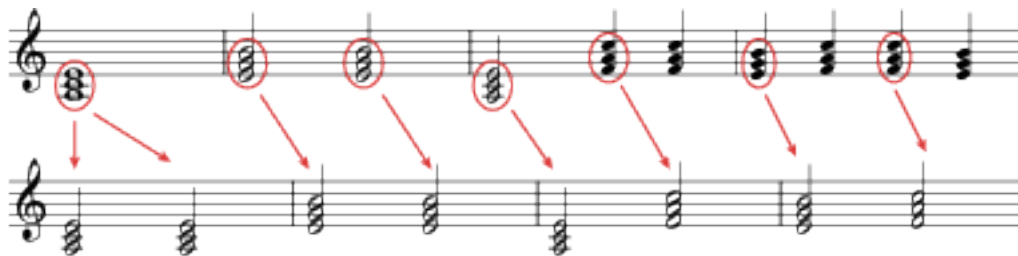
3.1.3 Melodi- och ackordanalys

Föregående rubrik beskrev en MIDI-fils uppbyggnad. Här presenteras hur denna information används av datorprogrammet för att extrahera melodi och ackord ur databasen.

Det första steget är att analysera melodin genom att plocka ut tonernas MIDI-händelser. Ur MIDI-händelsernas snabbmeddelanden hämtas tonernas MIDI-nummer som sedan konverteras till motsvarande toner. Tonernas längder beräknas som differensen mellan två händelsers positioner, den aktuella händelsen och den efterkommande. MIDI-nummer samt motsvarande längd sparades sedan för varje not i melodin.

Extrahering av ackord utförs på samma sätt som för melodin men med ett par skillnader. Eftersom ett ackord representeras av flera toner som spelas samtidigt behövs samtliga toners MIDI-nummer plockas ut och kopplas till motsvarande ackord. Beräkningen av ackordens längd var densamma som för melodin. Ackord med dess motsvarande längd sparas sedan undan för varje ackord i ackordföljden. Till skillnad från melodin utförs ytterligare behandling av ackordföljderna. Ackordens längd anpassas så att varje ackord tillhör en halvtakt. I figur 7 illustreras hur denna anpassning utförs för olika fall. Exempelvis delas ett ackord, som är en hel takt lång, upp i två halva takter istället. Detta görs för att lättare kunna koppla samman ackorden med melodin.

För att koppla samman ackord och melodi kapas melodin också vid varje halvtakt. Den associeras sedan med ackordet i samma halvtakt och datan skickas därefter vidare till modellerna för träning.



Figur 7: Olika fall för hur ackordföljden delas upp i halvtakter. Exempelvis delas ett ackord, som är en hel takt lång, upp i två halva takter istället. Förekommer fler än två ackord i takten förlängs det ackord som varje halvtakt inleds med.

3.1.4 Strukturanalys

I teoridelen av rapporten, avsnitt 2.1, användes Katy Perrys låt "Last Friday Night" som exempel på vad struktur i musik är. Dessa strukturer analyseras av programmet för att få mer självlikformighet i musiken.

Rapporten *Algorithmic compositions based on discovered musical patterns* av M. K. Shan och S. C. Chiu [24] presenterar en metod för att undersöka strukturen i barnvisor. I rapporten delar författarna in melodifraser i så kallade motiv som de sedan analyserar. En liknande metod används av programmet vid strukturanalysen och i denna rapport definieras ett motiv som en melodislinga med längd en takt.

Motiv kan även vara variationer av varandra. I exemplet går det att se att repetition är en vanlig motivförändring. Det visade sig efter analys av motiven i databasen att repetition var den överlägset mest förekommande motivförändringen. Programmet analyserar därför endast olika nivåer av repetition och dessa är följande

Motivförändring 1: 100 % repetition, om motivet upprepas som ett tidigare motiv med 100 % likhet,

Motivförändring 2: 75 % repetition, om motivet upprepas som ett tidigare motiv med 75 % - 100 % likhet,

Motivförändring 3: 50 % repetition, om motivet upprepas som ett tidigare motiv med 50 % - 75 % likhet.

Analysen av motiv och motivförändringar i ett musikstycke i databasen utförs genom att jämföra samtliga takter i musikstycket med varandra. Jämförelsen sker för varje sextondelsslag i takterna och utförs enligt

$$m_i = m_j \quad \text{om} \quad \frac{\sum_{s=1}^{16} \mathbb{I}\{n_i^s = n_j^s\}}{16} > P_{\text{limit}} \quad (7)$$

där m_i och m_j är de två motiven som jämförs. n_i^s och n_j^s är tonerna som spelas vid sextondel s i de båda motiven, $\mathbb{I}$ är en indikatorfunktion och P_{limit} är någon av de tidigare nämnda intervallgränserna. Informationen om motiven och de förändringar som hittas används sedan vid träning av modellen vilket beskrivs i avsnitt 3.2.1.

Programmet gör även en analys av återkommande sektioner eftersom det inom popmusiken är vanligt att repetition av takter om fyra görs, se tidigare exempel. Även denna information används sedan vid träningen av modellen för strukturgenerering.

3.1.5 Rytmanalys

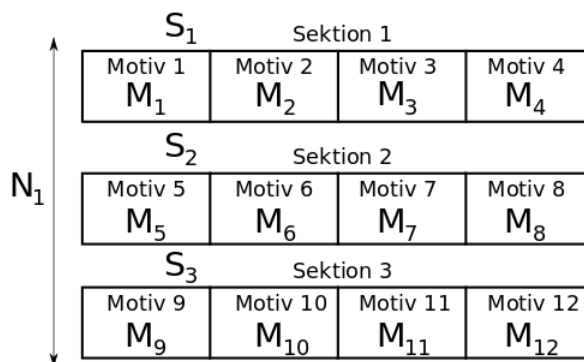
För att ge variation till den genererade musiken gör programmet en enkel analys av de rytmer som ackorden har i databasen. Rytmen analyseras för de första fyra takterna i varje låt. Därefter görs ett slumpval bland rytmerna som det genererade verket använder.

3.2 Modell och träning

I följande avsnitt beskrivs de maskininlärningsmetoder som utnyttjas av programmet, samt hur musiken modellerades. Avsnittet är uppdelat i tre delar: struktur, ackord och notföljder.

3.2.1 Modell för struktur

Strukturmodellen är den övergripande modellen för ett musikstycke och beskriver hur olika delar i musikstycket hänger ihop. Den struktur som framförallt förekommer i popmusik är återupprepning av motiv. Strukturen är uppbyggd av sektioner som i sin tur är uppbyggd av motiv enligt figur 8. Musikstyckena i databasen består alltid av två eller fler sektioner och programmet modellerades sedan utifrån detta. Viktigt att poängtera är att denna metod endast genererar en mall för följden av motiv och har ingenting med genereringen av själva notföljderna att göra.



Figur 8: Figuren illustrerar hur en struktur ser ut och hur den representeras med stokastiska variabler för fallet med tre sektioner. Varje sektion består av fyra motiv. Det finns tre olika sorters stokastiska variabler i figuren: en som beskriver antalet sektioner N , en som beskriver hur lika sektionerna är varandra S samt en som beskriver hur lika motiven är i förhållande till varandra M .

För att en struktur ska kunna genereras måste algoritmen tränas på flera olika sätt, med träningsdata. Denna data tillhandahålls med hjälp av strukturanalysen, beskriven i sektion 3.1.4, av databasen presenterad i sektion 3.1.1. Olika sätt av träning krävs för genomförandet av de olika stegen i genereringen. Dessa kan representeras med de stokastiska variablerna, N , S_n och M_n , enligt figur 8. N är antalet sektioner, S_n är sektion n och M_n är motiv n .

Den första stokastiska variabeln N har ett utfallsrum med tre element $\{2, 3, 4\}$ vilket innebär att antingen 2, 3 eller 4 sektioner bildas. Nästa stokastiska variabel S_n talar om hur lik sektion n ska vara en annan sektion. Den beskriver om motiv x i respektive sektion ska vara lika varandra. Däremot beskriver den inte eventuella likheter mellan motiv på olika platser i sektionerna, alltså för motiv x i den ena sektionen och motiv y i den andra om $x \neq y$ där $x, y \in \{1, 2, 3, 4\}$. Exempelvis kan den andra sektionens andra motiv vara identiskt med det andra motivet i den första sektionen, medan de resterande tre motiven i sektionerna skiljer

sig åt. Variabeln säger inte om till exempel första heltakten i första sektionen är lik tredje heltakten i andra sektionen. Den stokastiska variabeln S kan skrivas som

$$S = (x, y), x \in \{1, 2, 3, 4, 12, 13, 14, 23, 24, 34, 123, 124, 134, 234, 1234\}, y \in \{1, 2, \dots, k\},$$

där siffrornas värden i x anger vilka motiv som ska vara lika motiven i en annan sektion, y anger vilken sektion som den nya sektionen S ska likna och k är antalet sektioner som har genererats. Om till exempel $x = 24$ och $y = 1$ betyder det att sektion S ska vara lik den första genererade sektionen eftersom värdet på y är 1. Enligt värdet på x ska motiv två samt fyra i sektion S likna motiv två och fyra i den första genererade sektionen.

Den sista stokastiska variabeln M_n , se figur 8, används för att bestämma vilka andra motiv som ska vara lika mellan olika sektioner. Denna har utfallsrummet

$$\{0, 11, 12, 13, 21, 22, 23, \dots, n1, n2, n3\}, \quad (8)$$

där 0 innebär att ett helt nytt motiv, som inte liknar de tidigare motiven, ska genereras. För de resterade elementen i utfallsrummet gäller att första siffran talar om vilket av de tidigare motiven som det nya motivet ska likna. Den andra siffran motsvarar de motivförändringar som beskrevs i avsnittet om strukturanalysen 3.1.4. Till exempel innebär elementet 12 att det nya motivet ska vara likt det första motivet eftersom den första siffran för elementen är en etta. Den andra siffran i elementet innebär att motivet ska ha 75% likhet med det första motivet.

3.2.2 Modell för ackordföljder

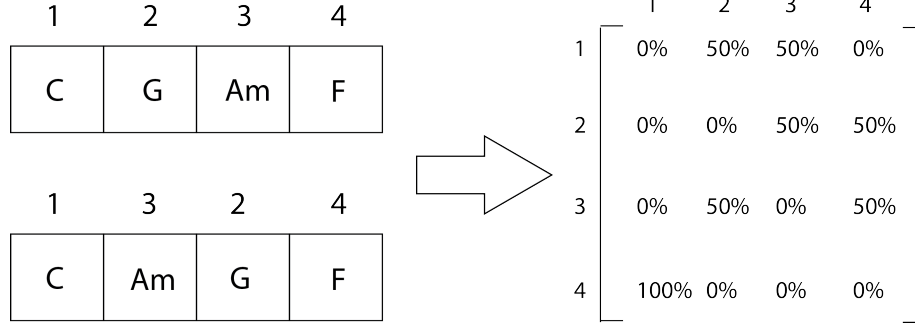
Vid generering av ackordföljder används en markovkedja av ordning $N = 3$. Som beskrivet i utvecklingsprocessen, se avsnitt 4.3, upptäcktes efter en övergripande undersökning av popmusik att det var vanligt att byta ackord en gång per takt. Därmed används en förenklad modell som speglar denna egenskap. Denna modell beskriver endast följden av ackord och inte rytmen som ackorden spelas med.

En händelse h i processen modellerades som ett ackord $0 \leq h \leq c_{max}$, där c_{max} är antalet unika ackord i träningsmängden $M = \{(h_n, h_{n+1}, h_{n+2}, \dots, h_{n_N})\}_n$. Träningsmängden består alltså av alla ackordföljder som innehåller $N+1$ ackord i databasen. Eftersom markovkedjan är homogen kan den beräknas i form av en övergångstensor

$$T_{\mathbf{h}} = \frac{|M_{\mathbf{h}}|}{|M_{\mathbf{h}'}|}, \quad (9)$$

med dimension $N + 1$, där $M_{\mathbf{h}} = \{\mathbf{h} : \mathbf{h} = (h_1, h_2, \dots, h_N), \mathbf{h} \in M\}$ och $M_{\mathbf{h}'} = \{\mathbf{h} : \mathbf{h} = (x, h_2, \dots, h_N), \mathbf{h} \in M, 0 \leq x \leq c_{max}\}$.

För att illustrera detta presenteras ett exempel i figur 9. Om en markovkedja av ordning ett, modellerad på tidigare beskrivet sätt, skulle tränas på ackordföljden i figuren skulle matrisen till höger i figuren skapas.



Figur 9: Ett exempel på en ackordföljd och dess motsvarande övergångstensor där ordningen på markovkedjan är ett. Eftersom ordningen är ett blir tensorn tvådimensionell, alltså en matris. Notera att matrisen har egenskapen beskriven i ekvation 4,

Eftersom detta är en markovkedja av ordning ett blir övergångstensorn tvådimensionell, det vill säga en övergångsmatris. Varje element i matrisen motsvarar därmed sannolikheten $P(h_{t+1}|h_t)$, där h_t är det föregående ackordet och h_{t+1} är nästkommande ackord. Denna matris kan sedan användas för att generera en ny ackordföljd.

För de första ackorden i en ackordföljd kan inte en markovkedja av ordning $N = 3$ användas eftersom det inte finns någon tidigare händelse att observera. Istället används kedjor av ordning $0, 1, 2, \dots, N - 1$ på ett sådant sätt att en modell av maximal ordning alltid används.

För stora värden på c_{max} och N blir tensorn i ekvation (9) snabbt för stor eftersom storlekskomplexiteten är $O(c_{max}^{N+1})$. Detta kan ge minnesproblem vid träningen men kan undvikas genom att använda andra modeller eller genom att träna mindre bitar av matrisen, något som inte gjordes för detta program.

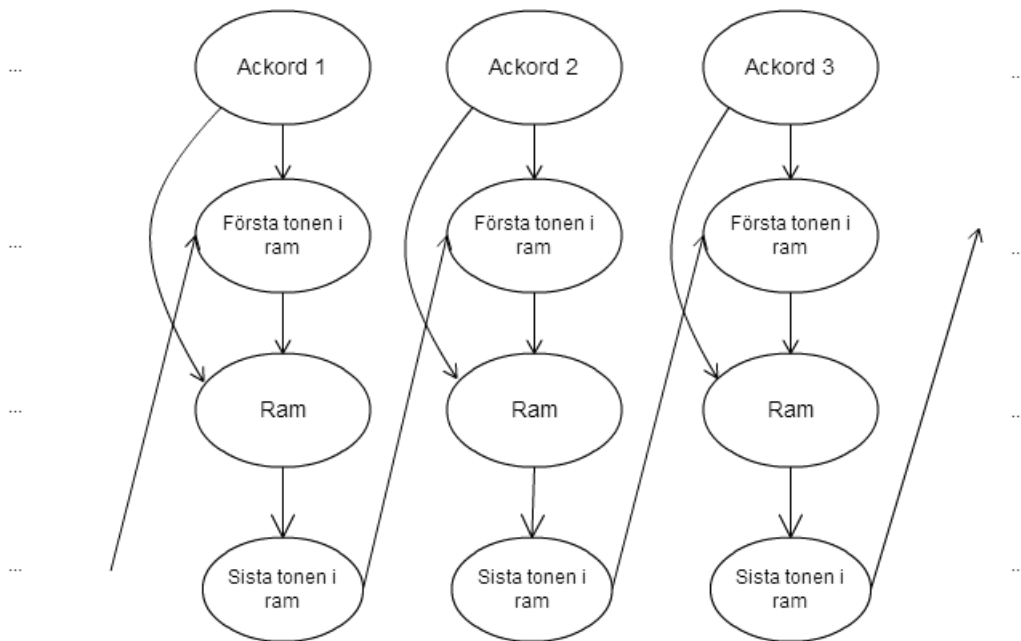
3.2.3 Modell för notföljder

För att förenkla uppgiften att generera notföljder är en låt modellerad som en sekvens av så kallade "ramar". En ram är en melodi som är en halvtakt lång och som spelas under ett ackord. Denna förenkling löste flera besvärliga problem, till exempel att få musiken harmonisk med ackorden. Olyckligtvis medför denna förenkling att en melodi aldrig binder över till nästa ram, vilket förekommer i popmusik. Att definiera en ram till en halvtakt lång snarare än en heltakt lång var ett val som gjordes för att spara minne och tid.

Den metod som användes för att generera notföljder givet en ackordföljd beskrivs som ett dynamiskt bayesiskt nätverk i figur 10. Pilarna i figuren beskriver beroenden och därmed även ordningen på genereringen. Den första tonen i en ram beror på den sista tonen i föregående ram och det tillhörande ackordet. Därmed genereras den första tonen i en ny ram först i notgenereringsprocessen. Utifrån den första noten och ackordet genereras sedan en melodislinga som fyller ramen. Ackordberoendet är också viktigt för att noterna ska passa till ackordet.

För att åstadkomma den tidigare beskrivna genereringsstrategin användes två markovliknande modeller. En av dem är en modell M bestående av tre stokastiska variabler $(\mathbf{p}_{new}, \mathbf{p}_{old}, \mathbf{c}_{new})$. Modellen beskriver sannolikheten att nästa ram i en sekvens av ramar ska börja på ton $\mathbf{p}_{new} < p_{max}$, givet att den tidigare ramen slutade på ton $\mathbf{p}_{old} < p_{max}$ och det nya ackordet $\mathbf{c}_{new} < c_{max}$. Där p_{max} är antalet olika toner som förekommer i databasen. Denna modell kan beräknas i form av en övergångstensor $T_{i,j,k} = P((\mathbf{p}_{new} = i) | (\mathbf{p}_{old} = j), (\mathbf{c}_{new} = k))$ enligt

$$T_{i,j,k} = \frac{|M_{ijk}|}{|M_{jk}|} \quad (10)$$



Figur 10: Ett dynamiskt bayesiskt nätverk som illustrerar strategin för att generera notföljder. Pilarna visar beroenden mellan olika delar och illustrerar därmed även ordningen som de olika delarna genereras i.

där $M_{ijk} = \{\mathbf{h} : \mathbf{h} = (i, j, k), \mathbf{h} \in M\}$ och $M_{jk} = \{\mathbf{h} : \mathbf{h} = (x, j, k), \mathbf{h} \in M, x \in \{0, \dots, p_{max}\}\}$. Träningsmängden har formen $M = \{(p_{new_n}, p_{old_n}, c_{new_n})\}_{n=1}^m$ och m är antalet datapunkter insamlade från låtarna i databasen.

Den andra modellen är en c_{max} lång lista av markovkedjor av ordning 2, tränade endast på notföljder som förekommer inom sitt eget ackord.

I dessa kedjor är varje händelse h i den stokastiska processen modellerad som en kombination av längd d och not p , $h = (d, p)$ där $p \leq p_{max}$ och $d \leq d_{max}$. p_{max} och d_{max} definieras till så små värden som möjligt givet träningsmängden. Denna modell beskriver alltså sannolikheten att spela nästa not givet de tre tidigare noterna och ackordet som den befinner sig i. I övrigt tränas dessa kedjor på samma sätt som markovkedjorna som genererade ackord, se (9). Detta resulterar i en tensor av storlek $(c_{max} * p_{max} * d_{max})^{N+1}$ vilket orsakar stora minnesproblem för stora värden på N . För att undgå detta problem skapas och tränas varje markovkedja vid behov.

3.3 Musikgenerering

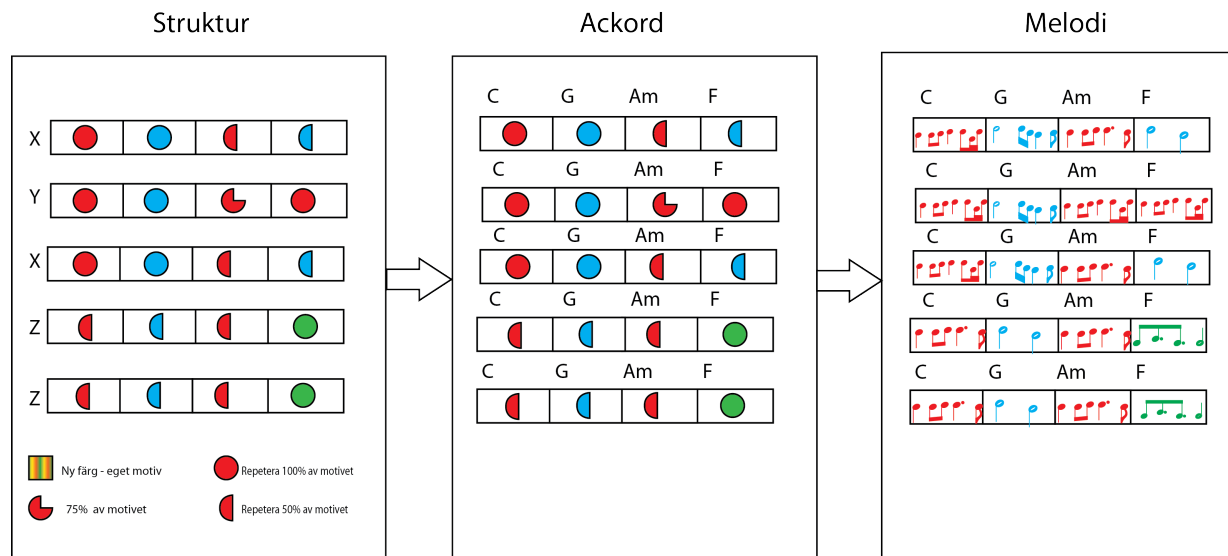
Musikgenereringen är uppdelad i tre delar: struktur, ackord och melodi, vilket representeras i figur 11.

Först genererar programmet strukturen, vilket sker i tre steg. Till att börja med genereras antalet sektioner som en musikdel ska bestå av. Därefter genereras sektionslikheterna och till sist genereras vilka motiv och motivförändringar som ska göras.

Efter strukturen genereras ackordföljder i form av fyra, en takt långa, ackord. Dessa följer återupprepas sedan under ett helt parti. Denna lösning valdes eftersom en översyn av popmusik visade att det var mycket vanligt. Generering sker med hjälp av övergångstensorn beskriven i ekvation (9).

Notföljder genereras sedan med hjälp av modellerna beskrivna i avsnitt 3.2.3. Om det enligt strukturmallen ska genereras ett nytt motiv genereras två nya ramar för att fylla takten. Om det istället ska ske en motivförändring kopieras de första 50% , 75% eller 100% av noterna i ramarna från originalmotivet och de

övriga genereras på nytt, se figur 3.3. Denna lösning är begränsad på det sättet att den endast kan ändra på slutet av ett motiv, vilket gjorde problemet med att göra motivförändringar simplare.



Figur 11: Figuren beskriver genereringstrategin som programmet använder sig av. Först genererar programmet en övergripande struktur för musikstycket. Varje takt får information om den antingen kommer innehålla ett nytt motiv eller om det kommer vara en förändring av ett tidigare motiv. Därefter genereras en ackordföljd med fyra ackord som upprepas. Slutligen genereras en melodi utifrån den tidigare informationen som skapades i strukturgenereringen.

3.4 Utvärdering av programkomponerad musik

Det kan vara svårt att bedöma musik objektivt eftersom upplevelsen skiljer sig från person till person. För att utvärdera musiken som programmet skapade gjordes därför två olika undersökningar. Dessa syftade inte till att bedöma kvaliteten på musiken utan om programmet kan skapa popmusik. Först gjordes en statistisk analys av låtarna i databasen där förekomsten av ackord, vanliga ackordföljder och strukturen undersöktes. Sedan jämfördes detta med en likadan undersökning av låtar som genererats. Den andra undersökningen som gjordes var en lyssnarundersökning i form av en variant på Alan Turings test [25]. En försöksgrupp fick lyssna på en mängd låtar, både datorgenererade och människokomponerade, och avgöra om dessa var komponerade av en människa eller av programmet. I detta avsnitt presenteras hur dessa försök genomfördes.

3.4.1 Statistisk analys av popmusik

Tre olika statistiska analyser genomfördes på databasen och de genererade låtarna. Först undersöktes vilka ackord som förekom oftast i låtarna i databasen och därefter beräknades sannolikheten för förekomsten av de tio vanligaste ackorden enligt

$$P(C) = \frac{N_C}{N},$$

där C är ackordet, N_C är antalet musikstycken som innehåller ackordet och N är antalet musikstycken som undersökts.

Vidare undersöktes även sannolikheten för att ett visst ackord följer ett annat. De tio vanligaste ackorden som följer ett C-dur användes. Sannolikheten för att ett av dessa ackord följde C-dur beräknades som

$$P(C_{t+1}|C_t) = \frac{N_{C_{t+1}}}{\sum_n N_{C_{t+1_n}}}$$

där C_{t+1} är det kommande ackordet, C_t är det spelade C-duret och $N_{C_{t+1}}$ är antalet gånger C_{t+1} följer C_t . $\sum_n N_{C_{t+1_n}}$ är summan av antalet ackord som följer C-duret. Samma beräkning utfördes sedan för de resterande ackorden så att ett diagram med sannolikheter för olika ackordövergångar kunde ritas.

Till sist undersöktes sannolikheten för förekomst av olika strukturer. Denna analys bestod av beräkning av sannolikhet för förekomst av olika typer av motivförändringar samt sannolikhet för vilka takter i de två första sektionerna av varje musikstycke som innehåller en motivförändring.

Sannolikhetsberäkningen för förekomsten av olika motivförändringar utfördes enligt

$$P(m) = \frac{n_m}{N} \quad (11)$$

där n_m är antalet låtar innehållande motivförändring m och N är antalet analyserade låtar.

Sannolikheten för vilka takter i de två första sektionerna som innehåller en motivförändring beräknades enligt

$$P = \frac{\sum_{k=0}^N \mathbb{I}\{m_i^k = m_j^k\}}{N} \quad (12)$$

där N är antalet analyserade låtar, m symboliserar ett motiv och $k \in [1, N]$. $i, j \in [1, 8]$ eftersom det finns fyra motiv i varje sektion. Villkoret $m_i^k = m_j^k$ är uppfyllt då motiven m_i^k och m_j^k är mer än 50% lika. Beräkningsmetoden för två motivs likhet presenterades i sektion 3.1.4 i ekvation 7.

I undersökningen analyserades 250 genererade musikstycken och 350 låtar från databasen.

3.4.2 Lyssnarundersökning

Lyssnarundersökningen skickades ut som ett webbformulär där testpersonerna fick lyssna till nio olika musikstycken där fyra av dessa var skapade av människor och fem var datorgenererade. De genererade låtarna valdes ut på förhand bland 250 genererade låtar och de människokomponerade låtarna valdes slumpmässigt ut från databasens poplåtar. Däremot slumpades det om då låten ansågs vara alltför känd, med målet att minimera antalet personer som hört poplåtarna förut.

Undersökningen inleddes med en introduktion till testet samt en exempellåt på MIDI-format. Detta exempel hade som avsikt att förbereda testpersonerna för MIDI-formatet för att minska formatets påverkan på resultatet. Därefter utfördes lyssnartestet genom att spela de nio låtarna i en på förhand slumpad ordning. Innan testet gick vidare till nästa låt fick testpersonen välja mellan svarsalternativen "Människa", "Dator" och "Jag har hört låten förut". Det var inte möjligt att gå tillbaka och ändra val av tidigare låtar. Slutligen fick testpersonerna möjligheten att lämna en kommentar till hur de tänkt under utförandet av testet och ange eventuella brister de identifierat hos den datorgenererade musiken.

Resultatet från lyssnarundersökningen analyserades och visualiserades på tre olika sätt. Först beräknades testpersonernas individuella resultat på frågorna enligt

$$R_p = \frac{n_r}{n_{tot} - n_h} \quad (13)$$

där R_p är resultatet för en testperson, n_r är antalet rätta svar, n_{tot} är totala antalet låtar i testet och n_h är antalet låtar som testpersonen redan hört tidigare. Om testpersonen hört låtar tidigare var dessa svar ointressanta för resultatet och behövde därför subtraheras från det totala antalet låtar för att beräkningen skulle vara korrekt.

Vidare analyserades förhållandet mellan svarsalternativen för varje låt. Denna analys hade som avsikt att ge information om fördelningen av svarsalternativ för varje enskild låt.

Slutligen analyserades fördelningen av valen "människa" och "dator" för varje testperson enligt

$$P_m = \frac{n_m}{n_{tot} - n_h} \quad (14)$$

och

$$P_d = 1 - P_m, \quad (15)$$

där P_m är den procentuella andelen val av "människa", n_m är antalet val av "människa" och P_d är den procentuella andelen val av "dator". n_{tot} och n_h har samma innebörd som i ekvation (13). Denna analys hade som avsikt att ge en uppfattning om spridningen av svarsalternativen och hur vanligt det var att testpersonerna uppfattade samtliga låtar som datorgenererade eller tvärtom.

4 Utvecklingsprocess

Metoden som programmet använder sig av för att generera popmusik är framtagen efter en längre utvecklingsprocess. För att motivera några av de metodval som gjorts och för att ge en bredare förståelse för hur programmet skapats presenteras därför i detta avsnitt utvecklingsprocessen för programmet.

Utvecklingen startade med en enkel markovkedja, med andra ord en markovkedja av ordning ett. Denna metod gav icke harmoniska melodier vilket medförde att en markovkedja av högre ordning testades. När tillräckligt harmoniska kompositioner erhöles genererades även ackord till melodierna.

4.1 Generering av melodi med enkel markovkedja

Det första steget i utvecklingsprocessen var att generera en melodi genom att använda en första ordningens markovkedja. En melodi som genererats på detta sätt presenteras i figur 12. Samtliga noter i melodin ligger i C-durskalan, vilken är den skalan som används i databasen. Majoriteten av notlängderna som finns i databasen finns även i den genererade melodislingan. De vanligaste är dock fjärdedelar, punkterade fjärdedelar och åttondelar. Detta kan jämföras med exempellåten i figur 2 som enbart innehåller dessa notlängder.

Ibland blir en följd av noter disharmoniskt placerade vid generering med en enkel markovkedja. I takt sju i den genererade melodislingan, markerat med blått, är notföljden mycket svårsläst. Detta beror på att en enkel markovkedja inte tar hänsyn till var i takten noterna skapas. Oftast beror detta på att endast en sextondel har skapats när det musikteoretiskt borde skapats två sextondelar. Detta medför att resterande toner förskjuts vilket vidare medför att melodin låter opassande för popmusik.



Figur 12: Figuren visar en melodi som är genererad med en enkel markovkedja. Notföljderna som är markerade i figuren visar på musikstycken som är disharmoniska. I det grönmarkerade området är skillnaden i nothöjden oharmonisk, medan i det blåmarkerade området är placeringen av noterna problemet.

En enkel markovkedja följer alltså inte alltid harmoniken av notföljder för pop. De grönmarkerade takterna i figur 12 är ytterligare exempel på detta. Dessa takter innehåller stora hopp som vanligtvis inte förekommer inom popmusik. Även detta kan jämföras med melodislingan i exempellåten.

Melodin saknar även självlikformighet, vilket brukar förekomma i popmusik.

Dessa brister medför att enkla markovkedjor inte är det bästa alternativet när melodier ska genereras. Genom att använda en markovkedja av högre ordning elimineras vissa av dessa problem.

4.2 Generering av melodi med markovkedja av högre ordning

I nästa steg genererades melodin med högre ordningens markovkedjor, vilket gav mer harmoniska notföljder, se figur 13. Notföljderna gör färre och kortare hopp än melodin genererad av en enkel markovkedja vilket gör att den passar bättre till melodigenerering. Det kan däremot fortfarande förekomma disharmoniskt placerade notföljder. En länk till musikstycket finns i referensdelen av rapporten [26].



Figur 13: Melodi som är genererad med en markovkedja av ordning 2. Melodin är mer harmonisk än i markovkedjor av ordning ett, se figur 12. Vissa av noterna är placerade oharmoniskt. Länk till musikstycket finns i referensdelen av rapporten [26]

Ett problem som uppstod när markovkedjor med hög ordning genererades var att matriserna, se avsnitt 2.2.1, som användes i programmet blev för stora. Detta ledde till oexekverbar kod eftersom programspråket som programmet skrevs i inte tillåter matriser av sådan storlek. Problemet försökte lösas genom att minimera antalet noter som analyserades samt genom att ta bort vissa notlängder. Däremot räckte inte denna lösning för ordning högre än tre.

Att öka ordningen på markovkedjan förbättrade inte självlikformigheten. Däremot gav markovkedjor tillräckligt bra resultat för att generera melodier. Istället utvecklades en annan metod för att ge musiken självlikformighet. Ett försök till att hitta en bättre metod för melodier gjordes med hierarkiska markovmodeller. Denna idé övergavs dock på grund av brist på vetenskapliga artiklar inom området.

4.3 Generering av melodi och ackord med dynamisk bayesiskt nätverk

Till en början var tanken att ackorden skulle genereras till notföljderna med hjälp av en dold markovmodell [27]. Efter vidare undersökning visade det sig att detta inte skulle skapa återupprepande ackordföljder. Istället gjordes valet att använda en markovkedja för att först generera ackord och sedan anpassa notföljderna till dessa. Först genererades en lång följd av ackord där varje ackord spelades i en halvtakt. Det visade sig dock vanligt inom popmusik att spela ett ackord per takt. Därför gjordes en förenkling där ett ackord per takt genererades för de första fyra takterna och sedan återupprepades denna ackordföljd för de kommande takterna.

För att lösa problemet med att tonerna i melodin försköts modellerades notföljder istället som en ram. Dessa genererades sedan efter varandra där den nya ramens första ton beror på den sista tonen i den förgående ramen. Notföljderna behövde också låta rena till de givna ackorden vilket gjorde en markovkedja

till en olämplig modell eftersom antalet händelser blev för många. Istället användes ett dynamiskt bayesiskt nätverk som stokastisk modell, vilket ledde till att färre händelser behandlas. Detta gjorde träning möjlig utan att problem med databrist uppstår.

För att ge musiken struktur och självlikformighet planerades en strukturanalys för att hitta vanliga notföljder i melodin så som upprepningar och slingor där tonerna är desamma men rytmen ändras. Metoden var tänkt att likna den som användes av M. K. Shan och S. C. Chiu[24] i rapporten "Algorithmic compositions based on discovered musical patterns". Det visade sig dock att notföljder som var vanliga i barnvisor, vilket var det som testades av M. K. Shan och S. C. Chiu[24], inte var lika vanliga i popmusik. Därför hade en implementering av detta inte haft någon större påverkan på resultatet. Istället gjordes en analys på repetitioner, något det fanns mycket av i popmusik och detta användes sedan vid generering.

5 Resultat

I detta avsnitt presenteras ett par utvalda musikstycken som genererats av programmet samt de likheter som finns med popmusik komponerad av människor. Två olika genererade verk har valts ut för att visa omfånget av de låtar som genererats. Vidare presenteras resultatet efter en lyssnarundersökning som utvärderar musiken som genereras av programmet.

5.1 Genererad musik: exempel på olika musikstycken som programmet genererat

Det datorgenererade musikstycket i figur 14a skulle, med resonemanget i teoridelen 2.1, falla bättre in i kategorin popmusik än det andra datorgenererade musikstycket i figur 14b. Detta är på grund av att vi i musikstycket kan se återkommande fraser med olika variationer. Dessa har markerats med gröna och blå rutor i figur 14a. Även ackordföljden är vanligare i popmusik eftersom alla ackorden hör till tonarten C.

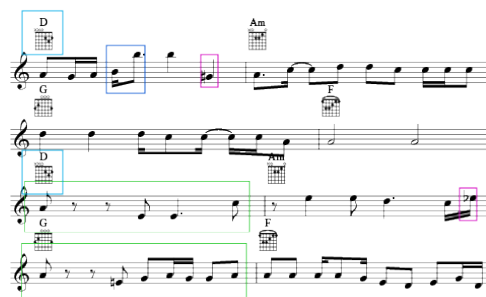
I det andra musikstycket ser vi istället skillnader från popmusik. Det första ackordet i ackordföljden är ett D, ett ackord som inte finns med i tonarten C. Då ett D spelas ökar risken för att toner som inte finns med i C-durskalan spelas, markerat med rosa rutor i figuren, vilket har hänt vid genereringen av detta stycke. Användningen av ett D ökar därmed risken för att musiken kommer att låta falsk. Ytterligare ett särdrag för det andra musikstycket är hoppet mellan toner, markerat i figuren med en mörkblå ruta. I popmusiken leder toner oftast till varandra, med andra ord följer varandra i skalan för att skapa ett sammanhang. Hoppet mellan tonerna bryter mot detta och därför låter musikstycket ”spretigt”. För att kunna lyssna på de genererade exemplen finns i referenslistan länkar till dessa [28][29].

Exempel 1 Genererad Låt



(a) En av låtarna som programmet komponerat. Detta exempel innehåller några av de delar som kännetecknar popmusik. Den ljusgröna rutan är en takt som upprepas med en viss förändring i takt två och fem. Även takt fyra återupprepas i takt åtta med en liten förändring. Tonererna i melodin ingår alla i C-durskalan och leder oftast till varandra. För att lyssna på detta stycke kan denna länk besökas [28].

Exempel 2 Genererad Låt



(b) En av låtarna som programmet komponerat. Detta exempel illustrerar hur programmet ibland genererar låtar som inte uppfyller teorin som musikteoridelen i sektion 2.1 beskriver. Det är få repetitioner, och vissa toner finns inte med i C-durskalan. Dessa är markerade med en rosa ruta. Även ackordföljden skiljer sig då den innehåller ackord som är ovanliga i tonarten C. För att lyssna på detta stycke kan denna länk besökas [29].

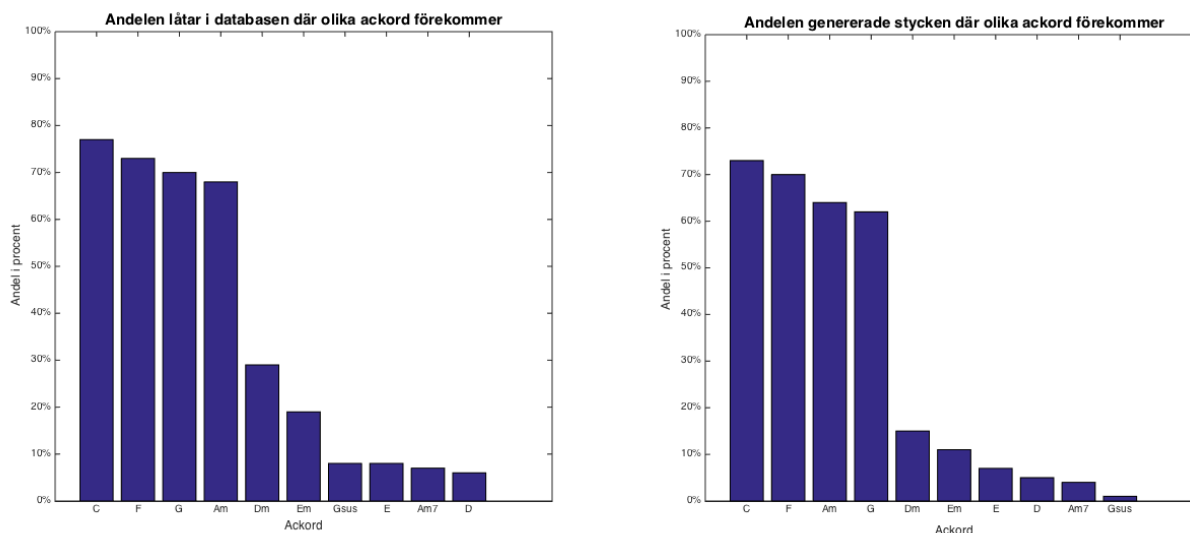
Figur 14: Två låtar genererade av programmet för att illustrera det omfång av musik som programmet genererar.

Datorprogrammet lyckades alltså generera verk i ett område som strukturmässigt och melodimässigt liknade popmusik, även om en del av musikstyckena hamnade utanför detta område. Något som programmet inte lyckades med var att generera krokar.

5.2 Statistisk jämförelse mellan databasen och den genererade musiken

I sektion 3.4.1 beskrevs utförandet av en statistisk analys av databasen och de genererade låtarna. Resultaten från denna analys kan användas för att jämföra de genererade låtarna med popmusik för att därmed kunna dra slutsatser om dess likheter.

Först analyserades hur ofta olika ackord dök upp i låtar och i figur 15 presenteras de tio vanligaste ackorden. Enligt figuren är de vanligaste ackorden, både för databasen och de genererade låtarna, C, F, Am och G. Detta är inte oväntat eftersom alla låtarna är transponerade till C och samtliga ackord finns med i tonarten.

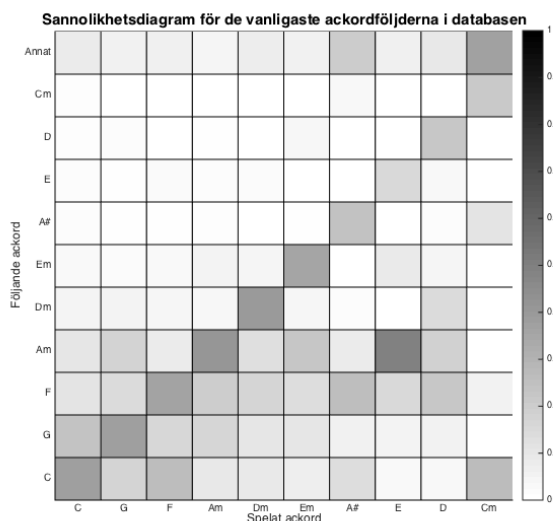


(a) De vanligast förekommande ackorden i databasens musikstycken. Eftersom låtarna alla gick i tonarten C är det inte förvånande att ungefär 75 % av alla låtarna innehöll detta ackord. Vidare är de kommande ackorden alla toner som ingår i C-durskalan fram till färgningen Gsus. För att skriva en poplåt i C bör låten innehålla ackorden C, F, G och Am.

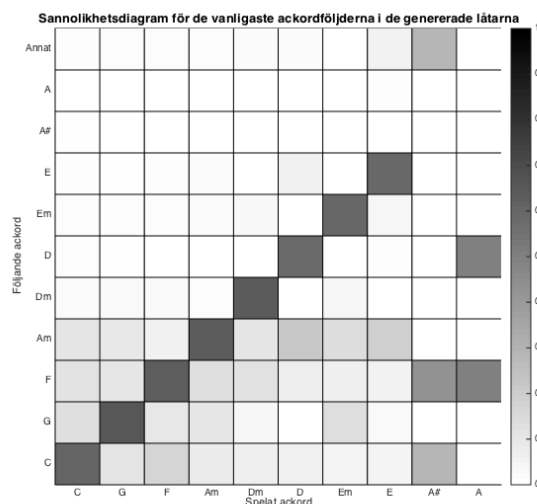
(b) I de genererade låtarna förekommer oftast ackorden C, F, Am och G. Dessa är även de vanligast förekommande ackorden i databasens låtar. Andelen skiljer sig dock från databasen för de mindre förekommande ackorden vilket kan bero på de förenklingar som programmet gör då den analyserar databasen.

Figur 15: I en undersökning av hur lik den datorgenererade musiken är popmusik gjordes en undersökning där de främst förekommande ackorden undersöktes. Det visade sig att samma ackord förekom oftast i den genererade musiken och i databasen.

Vidare undersöktes även vanliga ackordföljder och resultatet presenteras i figur 16. På horisontalaxeln visas det ackord som spelas, på vertikalaxeln visas det kommande ackordet och gråskalan representerar sannolikheten att det kommande ackordet följer det som spelas. Enligt diagrammen är det, både i databasen och i de genererade låtarna, vanligast att ett ackord följs av samma ackord igen. Detta är dock vanligare hos de genererade låtarna. För de fyra vanligaste ackorden, som visades i figur 15, är det vanligare att de olika ackorden följer varandra.



(a) De vanligaste ackordföljderna i databasen och sannolikheterna för dessa. Vanligt är att ackordet följs av samma ackord.



(b) De vanligaste ackordföljderna i de genererade låtarna och sannolikheterna för dessa. Bland de genererade låtarna är det vanligare att samma ackord följer sig självt än det är i databasen.

Figur 16: I de två figurerna presenteras de vanligaste ackorden att följa C i databasen och i de genererade låtarna. Sannolikheterna att dessa ackord följer varandra visas sedan i rutorna i diagrammet. Skalan presenteras bredvid de två diagrammen där mörkare betyder att det är vanligare att ackordbytet sker.

Slutligen gjordes en statistisk analys av strukturen i låtarna i databasen och i de genererade låtarna. Analysen gav statistik för de vanligaste motivförändringarna samt hur ofta det sker att hela sektioner repeteras. Analysen utfördes för musikstycken i databasen och för alla genererade låtar. Resultaten presenteras i tabellerna 1 och 2.

Tabell 1: Statistik för förekomst av motivförändringar i databasen. Den vanligaste motivförändringen är exakt repetition, därefter kommer motivförändring 3. Sannolikheten att en motivförändring överhuvudtaget förekommer i en låt är hög.

Motivförändringar i databasen	
Någon motivförändring	98 %
Motivförändring 1 (100 % repetition)	84 %
Motivförändring 2 (75-100 % repetition)	61 %
Motivförändring 3 (50-75 % repetition)	77 %
Repetition sektion	34 %

Tabell 2: Statistik för förekomst av motivförändringar i de genererade låtarna. Den vanligaste motivförändringen är motivförändring 3, därefter kommer exakt repetition. Sannolikheten att en motivförändring överhuvudtaget förekommer i en låt är hög.

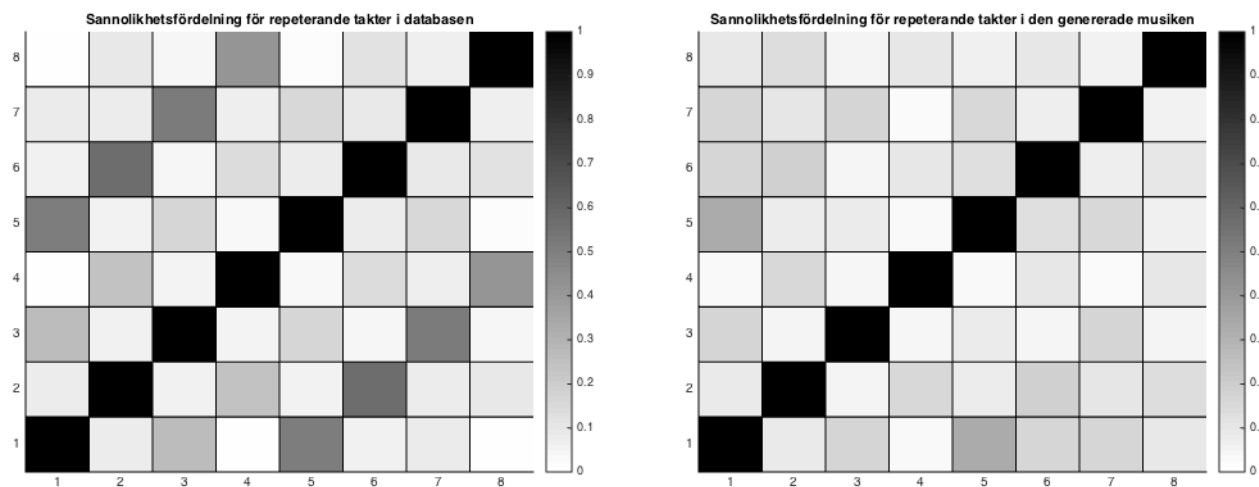
Motivförändringar i genererad musik	
Någon motivförändring	95 %
Motivförändring 1 (100 % repetition)	69 %
Motivförändring 2 (75-100 % repetition)	51 %
Motivförändring 3 (50-75 % repetition)	70 %
Repetition sektion	0.41 %

I tabell 1 går det att se att den vanligaste motivförändringen i databasens låtar är exakt repetition, och därefter kommer motivförändring 3. Sannolikheten att en motivförändring överhuvudtaget förekommer i en låt är 98 %. Eftersom samtliga motivförändringar är till en viss grad repetition kan slutsatsen dras att repetition av en takt långa melodislingor är vanligt i popmusik. Dessa värden kan vidare jämföras med motsvarande värden för den genererade musiken, se tabell 2. Bland de genererade låtarna förekom någon motivförändring i 95 % av samtliga låtar, ett resultat likt det för popmusiken. Ännu en likhet är att de

vanligast förekommande motivförändringarna är motivförändring 1 och 3, och motivförändring 2 är mindre vanlig. Vad som skiljer de genererade låtarna mest från databasens statistik är antalet repeterande sektioner. Bland de genererade låtarna är det bara 0.41 % som har två sektioner som liknar varandra, till skillnad från databasens 34 %.

Vidare utfördes även en analys av vilka takter som vanligast hade motivförändringar. I figur 17 presenteras sannolikhetsdiagram över detta. De svarta diagonalerna är ointressanta eftersom de endast förklarar att en takt med 100 % sannolikhet är lik sig själv. Däremot kan man se att taktparen 1-5, 2-6, 3-7 och 4-8 är avsevärt vanligare att repeteras hos låtarna i databasen. Däremot går det inte att påstå detsamma för de genererade låtarna, där det är en mer jämn fördelning bland de olika taktparen.

Om samtliga taktpar 1-5, 2-6, 3-7 och 4-8 innehåller motivförändringar motsvarar det att en sektion repeteras. Den markanta skillnaden är alltså samma som kan ses i tabellerna 1 och 2 för sektionsrepetitionen.



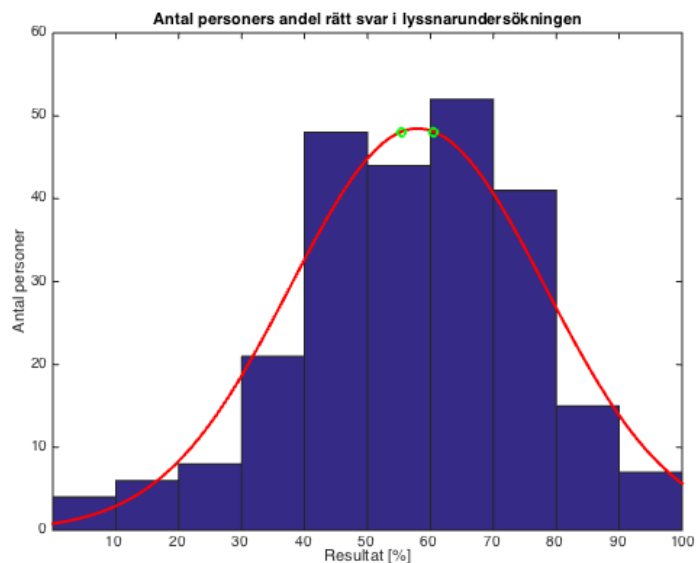
(a) Sannolikhetsfördelning för repeterande takter i databasen. Den svarta diagonalen är inte intressant eftersom den visar att en takt är till 100 % lik sig själv. De mörkgrå diagonalerna för taktparen 1-5, 2-6, 3-7 och 4-8 i sannolikhetsdiagrammet betyder att repetition av sektioner är vanligt.

(b) Sannolikhetsfördelning för repeterande takter i den genererade musiken. Den svarta diagonalen är inte intressant eftersom den visar att en takt är till 100 % lik sig själv. Taktparet 1-5 har en starkare nyans av grått relativt de övriga taktparen. Detta innebär att motivförändringar mellan takt 1 och 5 förekommer oftast.

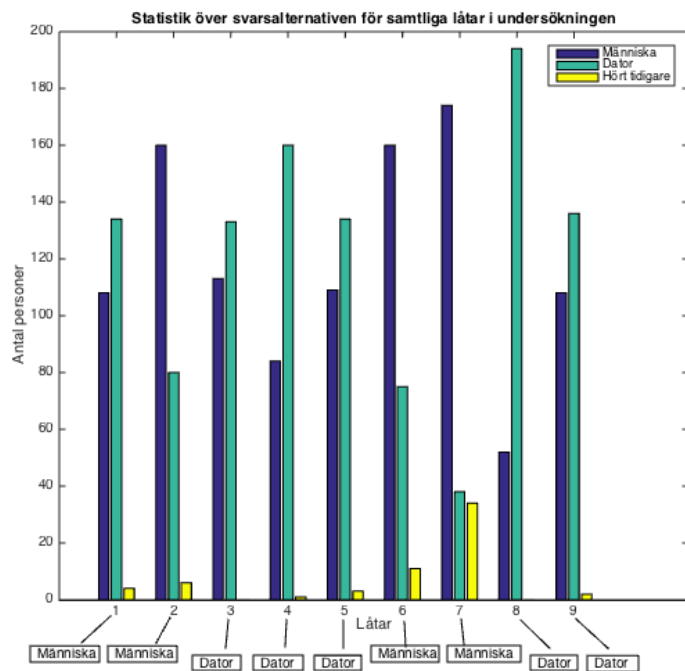
Figur 17: Sannolikhetsfördelning för motivförändring mellan takter hos de två första sektionerna i databasen och i de genererade låtarna.

5.3 Lyssnarundersökning: förhållandet mellan människo- och datorgenererad musik

Lyssnarundersökningen utfördes av totalt 246 personer och resultatet från undersökningen visualiseras i ett histogram i figur 18. Den röda linjen är en approximerad gausskurva och x-koordinaten till toppen av denna, $x \approx 58\%$, visar medelvärdet. Ett 95-procentigt konfidensintervalls gränser beräknades till 55 % och 61 %, vilket visas i figur 18 som gröna punkter. Om undersökningen skulle utföras oändligt många gånger skulle konfidensintervallet i 95 % av fallen innehålla det korrekta medelvärdet.



Figur 18: Resultat från undersökningen av genererade refrängstycken. Medelvärde av datan, 58 %, är x -koordinaten för toppen av den röda kurvan som är en gausskurva approximerad till datan. De gröna punkterna visar gränserna för det 95-procentiga konfidsintervallet.

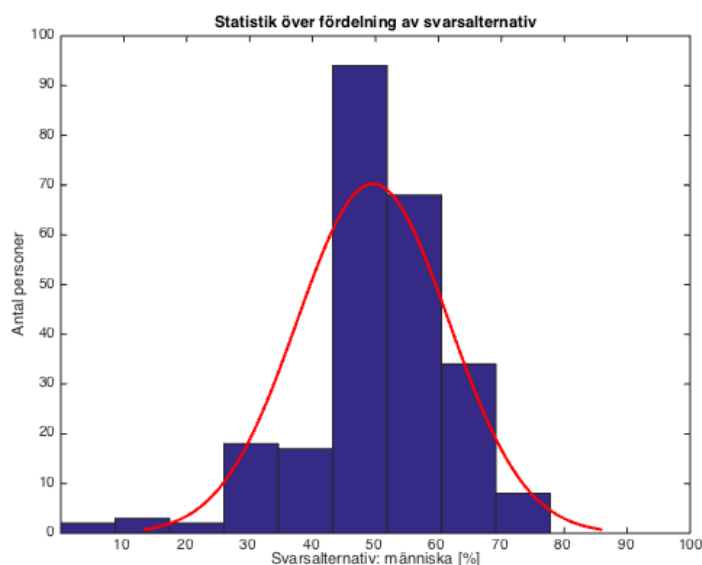


Figur 19: Antal val av de olika svarsalternativen för samtliga 9 musikstycken i undersökningen. Blå stolpar visar antalet testpersoner som valt "människa", gröna visar antalet val av "dator" och gul visar antalet som hört låten tidigare. Låt nummer 3, 4, 5, 8 och 9 är de datorgenererade låtarna. Eftersom de gröna stolparna för samtliga låtar är högst betyder att en majoritet lyssnare genomskådade att det var en dator som komponerade dem.

I figur 19 visas statistiken för de olika svarsalternativen för var och en av de nio låtarna. Låtnumren 3, 4, 5, 8 och 9 representerar de datorgenererade låtarna. Figuren visar exempelvis att låt nummer 3 var lättare att identifiera som datorgenererad än låt nummer 8. Låt nummer 3 representeras i figur 14a.

I figur 20 representeras data för fördelningen av valda svarsalternativ och man kan i den se att oftast väljer testpersonerna hälften av låtarna som datorgenererade och hälften som människokomponerade. Det hände aldrig att någon person valde samtliga låtar som människokomponerade, men i vissa fall har samtliga valts som datorgenererade.

I figuren kan man även urskilja en spridning åt höger vilket visar på att en stor andel testpersoner valt svarsalternativen, "människa" oftare än "dator". Notera att detta hände trots att testet innehöll fler datorkomponerade låtar än människokomponerade låtar.



Figur 20: Representation av fördelning av svarsalternativen. Ökande värden på x-axeln visar ökande andel val av "människa". Minskande värden på x-axeln visar därmed ökad andel val av "Dator". Vanligast är att testpersoner väljer hälften "människa" och hälften "dator".

Undersökningen innehöll även en möjlighet för testpersonerna att lämna kommentarer om musiken och ange eventuella brister de identifierat hos de datorgenererade låtarna. Kommentarererna var tvungna att ges innan testpersonerna fick reda på sitt resultat för att detta inte skulle påverka dem.

Vanliga kommentarer om de låtar som testpersonerna trodde var datorgenererade var "ryckig, ologisk melodi", "slumpmässigt", "konstiga notföljder" och "snabba toner". Genomsnittsresultatet för testpersonerna med dessa kommentarer var 60%.

Andra återupprepade kommentarer var att den datorgenererade musiken var "ej sångbar" och "svår att nynna till". Genomsnittsresultatet för testpersonerna med denna typ av kommentarer var 88% rätt.

6 Diskussion

Rapporten har beskrivit hur ett datorprogram som komponerar musik fungerar och den process som genomgåtts för att utveckla det. Olika modeller tränas på en databas för att sedan användas vid generering av ny musik. Det kan vara svårt att bedöma resultatet av ett sådant program eftersom vad som låter bra är en subjektiv åsikt. För att på ett objektivt sätt bedöma programmet försöker därför rapporten istället besvara frågan om programmet genererar popmusik till skillnad från om den genererar ”bra” musik. Detta gjordes med statistiska undersökningar som analyserade särdrag i popmusik och om samma särdrag kunde ses i den genererade musiken.

Musiken genererad av programmet lyckades fånga upp många aspekter av popmusik. Däremot misslyckades den med att fånga alla dessa egenskaper eftersom en majoritet av lyssnarna kunde höra skillnad på en människas verk och på programmets verk. Intressanta slutsatser kan däremot dras från de genererade musikstyckena och de modeller som använts. I de kommande delarna kommer därför resultat som erhållits och förbättringar av programmet att diskuteras under de kommande avsnitten.

6.1 Statistiska likheter och skillnader mellan genererade musikstycken och popmusik

För att utvärdera programmets förmåga att skriva popmusik undersöktes förekomsten av specifika ackord, ackordföljder samt strukturer. Undersökningen utfördes både på musiken i databasen och de genererade låtarna.

Resultatet pekade övergripande på att programmet lyckades skapa musik vars uppbyggnad efterliknade musiken i databasen. Figur 15 visar att det är störst sannolikhet att ett ackord följs av samma ackord både i de genererade låtarna och i databasens låtar. Denna likhet tyder på att programmet klarade av att välja ackord likt en popmusiker. Ytterligare analys av figur 15 visar att andelen vanligt förekommande ackord var ännu en likhet. Vad som däremot skiljde sig var andelen ovanligt förekommande ackord. Anledningen till detta är troligtvis den förenkling som programmet gör då den endast genererar en ackordföljd med fyra ackord som sedan upprepas. I databasen skulle till exempel en åtta takter lång ackordföljd kunna se ut som

C Am F C C Am F Dm.

Denna ackordföljd innehåller alltså ett av de mer ovanliga ackorden. Om de första fyra ackorden genereras med programmet kommer de att återupprepas och Dm-ackordet kommer därmed att falla bort. Detta leder till att färre Dm används i de genererade musikstyckena. Även förenklingen att ackorden analyserades i halvtakter kan ha påverkat förekomsten av vissa ackord eftersom en del ackord faller bort vid denna bearbetning av datan. Detta illustreras i figur 7.

Vanliga ackordföljder i låtarna undersöktes och resultatet presenterades i figurerna 16a och 16b. En liknande trend kan ses för de båda figurerna men det finns även viktiga skillnader att diskutera. Det visade sig att det är vanligast att ett ackord följs av samma ackord eftersom diagonalen i de båda bilderna är mörkast. Detta är inte oväntat eftersom ackordrytmen kommer att göra att det blir vanligare att samma ackord följer varandra, exempelvis när ackordet spelas på varje pulsslag i takten. Detta är dock vanligare i de genererade musikstyckena och detta beror troligtvis på samma förenkling som påverkade resultatet för förekomsten av ackord. Eftersom vissa ackord försvinner vid analysen av databasen ökar sannolikheten att samma ackord upprepas igen. Att ackord försvann ledde också till att sannolikheten att ett ackord följdes av ett ”annat” ackord blev nästintill noll.

En tredje undersökning av de genererade låtarna gjordes också där strukturen i musiken undersöktes. Även i detta fall överensstämde resultatet bra med de uppmätta värdena. Programmet klarar alltså att generera motivförändringar. Ett resultat som inte överensstämde var dock antalet repeterade sektioner. Även om

motiv repeteras i de genererade låtarna så är placeringen av dessa motivförändringar inte den samma som i databasens stycken. Detta beror troligen på modellvalet, men problemet hade kanske lösts genom att viktas sannolikheterna.

6.2 Lyssnarundersökning: förhållandet mellan människo- och datorgenererad musik

Under denna rubrik tolkas resultaten av lyssnarundersökningen. Diskussionen i denna del grundar sig på figurerna i resultatdelen.

Medelvärde av andelen rätt lyssnarna hade var 58 %, vilket kan ses i figur 18. Detta är relativt nära 50 % vilket betyder att det var svårt för en del testpersoner att urskilja de datorgenererade musikstyckena från de komponerade av en människa. Anledningen kan vara att den datorgenererade musiken liknar musik komponerad av människor. Om de genererade låtarna varit omöjliga att urskilja från låtarna skapade av människor borde konfidensintervallet omsluta 50 %. Avvikelsen på 5 % till 11 % tyder på att det finns något i de genererade låtarna som avslöjar att det är en dator som skapat dem.

Enligt resultatet i figur 19 trodde fler personer att de datorgenererade låtarna var datorgenererade än komponerad av en människa. Att majoriteten alltid svarat rätt på frågorna tyder på att det gick att genomskåda de datorgenererade musikstyckena. Något som lyssnarna märkte av under lyssnarundersökningen var att vissa musikstycken inte gick att sjunga och på så sätt insåg de att musikstycket var algoritmiskt komponerat. Troligtvis beror detta på kapningen av noter som ökar förekomsten av noter med kortare längder. Detta skulle kunna delvis lösas genom att låta en ram vara en heltakt lång, dock var inte det möjligt på grund av minnesproblem. En fullständig lösning skulle kunna vara att utveckla en metod som analyserar och modellerar sängbarheten i en låt.

I figur 20 ser man att lyssnarna valde alternativet människa mer än alternativet dator. Detta kan bero på att låtarna som skapas av den algoritmiska kompositören var svåra för lyssnarna att urskilja från de människokomponerade låtarna, vilket skulle kunna betyda att programmet skapar poplåtar som kan konkurrera med mänskliga låtskrivare. Resultatet kan dock också tolkas som att lyssnarna inte gillade någon av låtarna eftersom inga av lyssnarna svarade människa på alla frågor men vissa av dem svarade dator på samtliga frågor. Förklaringen till detta kan ligga i MIDI-formatet. Personer som lyssnar på popmusik är troligen vana vid att lyssna på andra format, som mp3-filer, där musiken får ett mer realistiskt ljud.

Det är viktigt att påpeka att vissa psykologiska faktorer kan ha påverkat resultatet av testet. Det är till exempel möjligt att lyssnarna främst jämförde låten de för tillfället lyssnade på med den föregående. För att minimera dessa effekter hade ordningen som låtarna presenterades i kunnat slumpas. Vidare togs inte hänsyn till testpersonernas miljö vid utförandet av undersökningen. Faktorer som kan ha påverkat resultatet är kvalitet av ljud, störningar från omgivning eller hjälp av andra personer. Dessutom påverkar det faktum att testpersonerna aldrig får någon återkoppling under tiden de utför undersökningen, vilket medför att de inte kan lära sig att höra skillnad på låtarna. Det är möjligt att personerna skulle få ett bättre resultat om de utfört testet igen.

6.3 Likheter och skillnader med tidigare studier

Inom algoritmisk komposition finns det flera sätt att modellera musik på än det som presenterats i denna rapport. I detta avsnitt diskuteras de skillnader och likheter som finns mellan dessa modeller och modellen som behandlas i denna rapport.

Något som inte fungerade så bra med programmet presenterat i denna rapport var strukturmodellen. En alternativ modell för att skapa strukturer i musiken är generativ grammatik. Denna modell används mycket

enligt Fernández och F. Vico[5] och har sitt ursprung ifrån grammatik i språk. Inom algoritmisk komposition används denna metod genom att sätta upp en uppsättning regler med syftet att från egenskaper på högre strukturnivå skapa detaljerade sekvenser av musikobjekt som representerar objekt i ett formellt språk. Musikobjekt genereras återkommande genom att använda olika former av omskrivningsregler i en sekvens härledningssteg vilket har mycket likhet med strukturmodellen i denna studie. Ett problem som uppstått med generativ grammatik är att trots att den större strukturen i musiken oftast fångas upp misslyckas modellen med att fånga det inre sammanhanget på detaljnivå.

Ett viktigt steg vid användning av generativ grammatik är hur uppsättningen av regler, som driver den generativa processen, ska aktiveras. I allmänhet är reglerna utformade i en hierarki som strukturmodellen i denna rapport. Det finns många sätt att aktivera regler med en generativ grammatik men något som är vanligt förekommande är användning av aktiveringssannolikheter för reglerna. I en generativ grammatik förekommer det oftast tydligare regler jämfört med rapportens presenterade modell vilket skulle förbättra resultat vid strukturgenerering. Den generativa grammatiken kan dock öka risken för att den genererade musiken blir enförmig eller i alltför hög grad kopiera musik som redan finns. Det skulle därför vara intressant att testa metoden och jämföra det nya resultatet med resultatet i denna rapport.

Det har gjorts liknade tester som lyssnarundersökningen i denna rapport men med generativ grammatik [30]. Resultaten blev relativt lika varandra, vilket tyder på att dessa modeller ger liknande effekter på musiken som genereras.

En metod som liknar generativ grammatik är regelbaserad inlärning som testats i tidigare studier [5]. I denna metod finns det förutom statiska regler även delar som dynamiskt kan ändras eller läras in. Leistikow[22] beskriver hur ett system använder musikdata för inlärning av regler genom att definiera musikteoretiska regler som bivillkor till optimeringsproblemet med att minimera entropin. För aktiveringen av de olika reglerna används dynamiska bayesiska nätverk samt olika typer av markovmodeller vilket har många likheter med det system som rapporten behandlar. Skillnaden ligger framförallt i inlärningen och resultatet som Leistikow[22] erhöll, vilket blev att melodierna fick mer sammanhang och tydligare teman återfanns genom musikstyckena. Dessutom gav Leistikow[22] lösning en mer anpassningsbar modell eftersom bivillkor i modellen enkelt kunde modifieras.

En rapport som nämnts tidigare i rapporten och som använt en liknande metod är ”Algorithmic compositions based on discovered musical patterns” av M. K. Shan och S. C. Chiu[24]. På samma sätt genererades en övergripande struktur på verket för att sedan fortsätta med detaljer. De gjorde även ett liknande lyssnartest, med skillnaden att testpersonerna fick rangordna musiken de lyssnade på. De fick ett bättre resultat, vilket förmodligen beror på alla de förenklingar som gjorts i programmet som denna rapport beskriver. Framförallt förenklingen att melodin delades in i motiv som jämfördes med varandra för att få ut strukturen. Deras bättre resultat kan också bero på att de endast behandlade barnvisor. I programmet utvecklat av M. K. Shan och S. C. Chiu[24] jämfördes alla möjliga likheter inom ett musikstycke vilket ökade självlikformigheten i den genererade musiken.

Något annat som skiljer de båda programmen åt är att programmet behandlat i denna rapport använder markovkedjor vid generering av ackordföljder medan det andra programmet genererar ackordföljder genom att samla in delmängder av förekommande ackordföljder från data. Eftersom ackordföljderna varken har mycket variation eller påverkar resultatet i samma grad som struktur och melodi, påverkade inte detta metodval resultatet.

Ett stort problem med programmet, som nämns i brister och förbättringar, var att musiken inte blev självlikformig nog. För att lösa detta problem skulle kanske en fraktalbaserad metod kunna användas. En studie som använder en sådan metod beskrivs i [31].

6.4 Övriga brister som programmet uppvisade och lösningar till dessa

De kommande styckena kommer att beskriva problem som försämrar programmets prestation samt hur de kan lösas. Vissa av dessa problem har endast lösts med förenklingar, som gav en negativ effekt, och kan förbättras.

Ett av de första problemen som stöttes på var minnesproblem. Detta berodde på att metoden var implementerad på ett minnesineffektivt sätt. Istället för att optimera koden gjordes förenklingar som hade negativ effekt på resultatet. Minneseffektiv kod hade gjort det möjligt att använda högre ordning på markovkedjorna och tillåtit fler längder på noterna.

En förenkling som användes i programmet var att noter som gick över halva takter kapades. Detta utfördes på grund av att inget bra sätt att binda över noter mellan takter hittades samt för att kunna koppla ihop noter och ackord. Problemet som skapades av detta var att det tillkom fler korta noter i datan vilket gjorde musikstyckena hackiga. Problemet är trivialt om det tidigare nämnda minnesproblemet löses.

Strukturgenereringen och motivförändringarna resulterade i verk som mer liknade poplåtar. Däremot verkar det, med tanke på de kommentarer som mottogs under lyssnarundersökningen, troligt att verken inte är repetitiva och självlikformiga nog. En förbättring som kan göras är att öka sannolikheten för motivförändringarna, vilket hade lett till fler repetitioner.

7 Slutsatser

Den här rapporten behandlar ett datorprogram som komponerar popmusik. Musiken som programmet skapar har många likheter med popmusik komponerad av mänskliga kompositörer men skiljde sig dock i musikstyckenas självlikformighet och sångbarhet. Därför kan programmet ej fungera som ersättning för dagens popkompositörer

Däremot skulle programmet kunna fungera som en inspirationskälla för kompositörer. Kompositörerna skulle alltså kunna använda programmet och lyssna på låtarna som skapas tills de hittar ett stycke som de gillar. De skulle sedan kunna skriva en låt baserat på vad de gillade i stycket.

Även om programmet som beskrivs i denna rapport i nuläget inte kan ersätta dagens popkompositörer skulle framtida projekt, inom samma kategori, möjligtvis kunna konkurrera med dem.

Referenser

- [1] R. Kurzweil, *How to create a mind: The secret of human thought revealed*. New York, New York: Penguin Group Inc., 2012.
- [2] L. A. Hiller och L. M. Isaacson, *Experimental Music: Composition with an electronic computer*. CT, Westport: Greenwood Publishing Group Inc., 1979.
- [3] J. Biles, “Genjam: A genetic algorithm for generating jazz solos,” in *Proceedings of the International Computer Music Conference*. International Computer Music Association, 1994, pp. 131–131.
- [4] J. P. Boon och O. Decroly, “Dynamical systems theory for music dynamics,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 5, no. 3, pp. 501–508, 1994.
- [5] J. D. Fernández och F. Vico, “AI Methods in Algorithmic Composition: A Comprehensive Survey,” *Journal of Artificial Intelligence Research*, pp. 513–582, 2013.
- [6] M. Simoni och R. B. Dannenberg, *Algorithmic Composition: A Guide to Composing Music with Nyquist*. Michigan: University of Michigan Press, 2013.
- [7] J. Maurer, “A brief history of algorithmic composition,” 1999.
- [8] R. M. Keller och D. R. Morrison, “A grammatical approach to automatic improvisation,” 2007.
- [9] S. Tipei, “Mp1: a computer program for music composition,” 1975.
- [10] P. P. Cruz-Alcázar och E. Vidal-Ruiz, “Learning regular grammars to model musical style: Comparing different coding schemes,” in *Grammatical Inference*, vol. 1433.
- [11] C. Thornton, *Hierarchical markov modeling for generative music*. Ann Arbor, MI: MPublishing, University of Michigan Library, 2009.
- [12] P. M. Todd och Dr. G. Loy, *Music and connectionism*. MIT Press, 1991.
- [13] K. Verbeurg et al., “A hybrid neural-markov approach for learning to compose music by example,” in *Advances in Artificial Intelligence*, vol. 5, no. 3. Claremont, California: Springer, 2004, pp. 480–484.
- [14] M. Marques et al., “Music composition using genetic evolutionary algorithms,” in *Proceedings of the 2000 Congress on Evolutionary Computation*, La Jolla, CA, 2000, pp. 714–719.
- [15] R. Harris och J. Fitch, “Algorithmic composition of jazz,” *Master’s thesis, University of Bath, Department of Computer Science*, 2008.
- [16] S. Frith et al., *The Cambridge companion to pop and rock*. Cambridge, UK: Cambridge University Press, 2001.
- [17] T. Warner, *Pop music: technology and creativity: Trevor Horn and the digital revolution*. Ashgate Publishing Ltd, 2003.
- [18] B. Ulin, *Matematik & Musik*. Gleerups, 2004.
- [19] A. Appleby, *You Can: write a song*. Music Sales, 1995.
- [20] H. Hjortek och K. G. Johansson, *Arrboken: arrangering i pop & rockstil*. Danderyd, Sweden: Notfabriken Music Publishing, 2013.
- [21] H. P. Hsu, *Probability, Random Variables and Random Processes*. NewYork, New York: McGraw-Hill Education, 2013.

- [22] R. J. Leistikow, “Bayesian modeling of musical expectations via maximum entropy stochastic grammars,” Ph.D. dissertation, Department of Music., Stanford University, 2006.
- [23] A. Ligeza, *Logical Foundations for Rule-Based Systems*. Springer, 2006.
- [24] M. K. Shan och S. C. Chiu, “Algorithmic compositions based on discovered musical patterns,” *Multi-media Tools and Applications*, vol. 46, no. 1, pp. 1–23, 2010.
- [25] A. Burdett et al., *BCS Glossary of Computing and ICT*. Swindon, England: British Informatics Society Limited, 2008.
- [26] A. Lorentzson, “Markov andra ordningen *Youtube*. 28 maj, 2015. online,” <https://youtu.be/q7e-YL6CLco>, hämtad: 28-05-2015.
- [27] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: The MIT Press, 2012.
- [28] A. Lorentzson, “Exempel 1 *Youtube*. 28 maj, 2015. online,” <https://youtu.be/f4hKVfAV87w>, hämtad: 28-05-2015.
- [29] —, “Exempel 2 *Youtube*. 28 maj, 2015. online,” <https://youtu.be/gUPHsk9e5q8>, hämtad: 28-05-2015.
- [30] H. A. G. Salas et al., “Automatic music composition with simple probabilistic generative grammars,” *Polibits*, vol. 44, pp. 59–65, 2011.
- [31] P. S. Langston, “Six techniques for algorithmic music composition,” in *15th International Computer Music Conference*. Morristown, New Jersey: Bellcore, 1989.