NTNU
Norwegian University of
Science and Technology

Using Hidden Markov Models for Musical Chord Prediction

Gunhild Elisabeth Berget

Master of Science in Physics and Mathematics
Submission date: July 2017
Supervisor: Jo Eidsvik, IMF

Norwegian University of Science and Technology
Department of Mathematical Sciences

Abstract

Music is made up of a melody and chords that accompany the melody. Finding suitable chords, can be hard and time consuming. This thesis investigates the use of Hidden Markov models (HMMs) to assist in this process. The idea is to learn the underlying Markov chain chord progressions from training data, and then adjust the chord progressions using melody input.

Four models are suggested and compared, all of them a type of HMM. Different definitions of state spaces and different orders of the models are used. Considering whole measures instead of single beats as our states, result in an improvement of the predictions. Improved predictions are also obtained by building separate models for minor key songs and major key songs. Higher order models do not improve the results. The best performing model obtains a score of 70% when using leave-one-out-cross-validation (LOOCV) for a training data set containing 64 children's songs.

Sammendrag

Musikk består som oftest av en melodi og av akkorder som akkompagnerer melodien. Det å finne passende akkorder kan være en vanskelig og tidkrevende prosess. Denne oppgaven undersøker om skjulte Markov modeller kan bli brukt som assistanse i denne prosessen. Ideen er å lære den underliggende Markov-kjeden i modellen akkordprogresjoner fra treningsdata, for så å justere denne modellen basert på observasjoner av melodien.

Fire modeller blir foreslått og implementert, hvorav alle er en variant av en skjult Markov modell. Det vil bli benyttet forskjellige tilstandsdefinisjoner og forskjellig orden på modellene. Ved å definere en tilstand som en hel takt i musikken, i stedet for kun et slag, oppnås det en forbedring i resultatene. Resultatene blir også bedre når separate modeller bygges for dur og moll. Høyere ordens modeller gir ingen nevneverdig forbedring av prediksjonene. Metodene blir evaluert ved hjelp av utelat-en-om-gangen kryssvalidering. For treningsdata bestående av 64 barnesanger får vi et resultat på 70% korrekte prediksjoner med den best presterende modellen.

Preface

This report is the result of my final work in my Master of Science degree in Physics and Mathematics at the Department of Mathematical Sciences (IMF) at the Norwegian University of Science and Technology (NTNU). The work has been done during my tenth semester at NTNU, spring 2017.

I would like to thank my supervisor, Jo Eidsvik, for letting me do this project and for excellent guidance. I would also like to thank Andreas Asheim for interesting discussions and useful feedback throughout the writing process.

Gunhild Elisabeth Berget
Trondheim, July 2017

Contents

List of Figures **vii**

1 Introduction **1**

2 Basic Music Theory **3**
2.1 Notes . . . 3
2.2 Representation . . . 4
2.3 Scales and key of songs . . . 6
2.4 Chords . . . 9
2.5 Chord progressions . . . 10
2.6 Music genres . . . 12

3 Hidden Markov Model **13**
3.1 Markov Chains . . . 13
3.2 Hidden Markov Models . . . 15
3.3 The likelihood model . . . 16
3.4 The forward-backward algorithm . . . 17
3.4.1 Forward recursion . . . 17
3.4.2 Backward recursion . . . 18
3.4.3 Pseudocode . . . 19
3.5 The Viterbi Algorithm . . . 20
3.6 Second-order HMM . . . 22

4 The models for musical chord prediction **25**
4.1 Beat1 . . . 25
4.2 Measure1 . . . 27
4.3 Measure1m . . . 28

4.4 Measure2 29
4.5 Overview of the models 30

5 Training data and implementation **31**
5.1 Training data 31
5.2 Implementation 33
5.2.1 Estimation of transition and emission matrices 33
5.2.2 Model validation 34

6 Results and discussion **35**
6.1 Transmission matrices 35
6.2 Validation scores 38
6.3 Distribution of predicted chords 39
6.4 Discussion of prediction error 41
6.5 Silent Night 43
6.6 Byssan Lull 45
6.7 Fagert er landet 47

7 Closing remarks **51**
7.1 Summary 51

A List of the songs used as training data **57**

List of Figures

2.1 Notes on the piano. 4
2.2 An excerpt from ”Silent Night” illustrating sheet music. 5
2.3 Pitch of notes in sheet music. 5
2.4 Duration of notes and rests in sheet music. 6
2.5 The scale of C major. 7
2.6 The scale of A minor. 7
2.7 Key signatures 8
2.8 ”Silent Night” transposed. 8
2.9 C major and A minor chord in sheet music. 9
2.10 C major and A minor chord on the piano. 9
2.11 Chords built on the scale of C major. 11
2.12 The harmonic A minor scale. 11

3.1 First order Markov chain. 14
3.2 First order HMM. 15
3.3 Second order HMM. 23
3.4 HMM with augmented states. 23

4.1 Illustration of Beat1. 26
4.2 Illustration of Measure1. 27
4.3 Illustration of Measure1m. 29
4.4 Illustration of Measure2. 30

5.1 Chord and note distribution of training data. 32
5.2 Major chord distribution and minor chord distribution. 33

6.1 Transition matrices 36

6.2 Reduced transition matrices. 37
6.3 Distributions of predicted chords. 40
6.4 Confusion matrices. 42
6.5 Sheet music representation of "Silent Night". 43
6.6 Prediction results for "Silent Night". 44
6.7 Sheet music representation of "Byssan lull". 45
6.8 Prediction results for "Byssan lull". 46
6.9 Sheet music representation of "Fagert er landet". 47
6.10 Prediction results for "Fagert er landet". 48

Chapter 1

Introduction

In this thesis, we want to see if machines and algorithms can help creating music. Most composers have heard a vast amount of music. They are generally not creating something out of the blue. They use patterns they have heard before, and compose based on prior experience. In addition to this, they follow some basic rules of harmonization. This thesis investigates the possibilities of capturing these regularities in music using statistical learning methods.

A typical musical piece consists of a melody and of chords that accompany the melody. The process of finding suitable chords is called harmonization, and can be both hard and time consuming. This thesis proposes a method that assists in this process. We consider the usage of Hidden Markov Models (HMMs) for predicting musical chord progressions given a melody input.

A HMM is a convenient tool for modelling double stochastic processes, where one of these processes is not observable (hidden). The methodology and theory behind HMMs are described in many sources, among others (Rabiner and Juang, 1986; Ghahramani, 2001; MacDonald and Zucchini, 1997; Ibe, 2013). The HMM is a widely used method especially within pattern recognition. Two important application areas are within speech recognition (Rabiner, 1989; Picone, 1990; Levinson, 1986) and biological sequence analysis (Krogh et al., 1994; Thompson, 1983).

When musicians assign chords, they consider the adjacent chords and the notes appearing in the melody. We want to exploit these dependencies in the music.

By considering the chords to be hidden variables in a HMM, the dependencies between chords can be modelled with a Markov chain. In a Bayesian setting, this can be considered as the prior knowledge. The notes in the melody are regarded as observations, and our prior belief is updated with this new information to obtain a posterior distribution. Lastly, this posterior distribution is used for prediction.

By varying the state space of the HMM and the order of the model, four methods are proposed. The models are trained using a data base of training data consisting of 64 lead sheets of children's songs commonly used in Norway. Leave-one-out-cross-validation (LOOCV) is used to evaluate the different models.

Similar problems, combining music with statistical methods have been considered. The article (Allan and Williams, 2005) uses a HMM for harmonization of chorales, (Suzuki and Kitahara, 2014) explores Bayesian network models that generate four-part harmonies and (Cunha and Famalho, 1999) combines a neural network with a rule-based approach for chord prediction. The article (Chen et al., 2015) compares multiple machine learning methods for our problem, among others, a simple HMM. The article has been an inspiration for this thesis, and we want to further develop the HMM used in this.

Chapter 2 gives an overview of basic music theory. This is needed to get a better understanding of the results. Chapter 3 discusses the theory behind HMMs, and the models used in this thesis are presented in Chapter 4. The training data and the implementation of the methods are explained in Chapter 5. Chapter 6 presents and discusses the results, before some closing remarks are given in Chapter 7.

Chapter 2

Basic Music Theory

This chapter presents a basic introduction to music theory. This is given to get a better understanding of the proposed models and the results we obtain.

2.1 Notes

Music can be considered as sound arranged in time. The sounds are notes, and at a single time step there can be one note sounding alone, or multiple notes sounding together. The pitch of a note is defined by the frequency of the sound wave. A high frequency leads to a high-pitched note, and low frequency leads to a low-pitched note. In western music, twelve note names are defined based on the frequency of the note. Table 2.1 shows note names and some of their most common corresponding frequencies.

Note	C	C♯	D	D♯	E	F	F♯	G	G♯	A	A♯	B
Freq (Hz)	130.8	138.6	146.8	155.6	164.8	174.6	185.0	196.0	207.7	220.0	233.1	246.9
Freq (Hz)	261.6	277.2	293.7	311.1	329.6	349.2	370.0	392.0	415.3	440.0	466.2	493.9

Table 2.1: Table showing the note names together with some of their most common corresponding frequencies.

As seen from the table, there are more than one frequency corresponding to one note. When the ratio of the frequency of two notes is equal to any integer power of two, they are perceived very similar when listening to them, and hence the two notes has the same note name. As an example, do both 220 Hz and 440 Hz, correspond to the note A, which have a 2:1 ratio. In this thesis, we use the twelve notes from Table 2.1 in our models. We do not distinguish between high pitched or low pitched notes of the same kind, because this does not affect the harmonization and our problem of chord prediction.

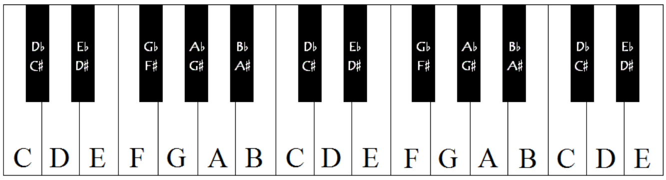


Figure 2.1: Illustration of a piano with the name of the notes written on the tangents.

Figure 2.1 shows the notes on a piano. Using the piano is an easy way to picture the notes and their relation, and it will be used as reference later in this chapter and throughout the thesis. The piano has the lower pitched notes to the left and the pitch increases as we go to the right. The white tangents represent the notes without accidentals ($\sharp$ and b) and the black tangents represent the notes with accidentals.

2.2 Representation

A common way of denoting music, is to use sheet music. Figure 2.2 shows an excerpt from the sheet music of "Silent Night". This song will be used as an example throughout the thesis. Some basic concepts are circled in the figure, and are explained below.

The set of horizontal lines is called the staff. The notes are drawn in the staff, and the pitch of a note is defined by where in the staff it is drawn. The higher up in the staff, the higher the pitch. Figure 2.3 gives an overview of the different pitches and the notes. A note can also be raised or lowered half a step by adding accidentals, where a $\sharp$ raises the pitch and a b lowers the pitch.

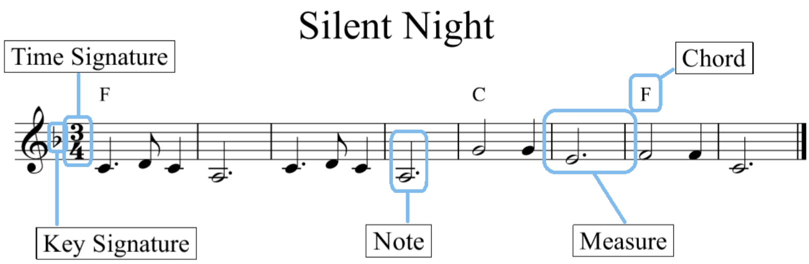



Figure 2.2: An excerpt from "Silent Night" illustrating some basic concepts of sheet music.

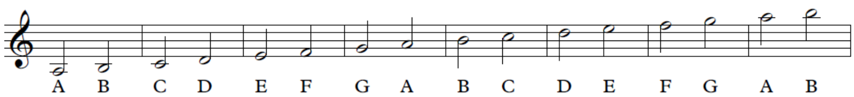



Figure 2.3: Illustration of the pitch of the notes in sheet music.

Apart from knowing the pitch of a note, a musician must know how long to play the note. The duration of a note is defined by the shape of the note. Figure 2.4 gives an overview of some different types of notes and their duration. The figure also displays the equivalent rests; the symbols used when there is a pause and no sound in the melody.

Furthermore, the songs are divided into bars or measures by vertical bar lines. The length of a measure is defined by the time signature. In Figure 2.2 the time signature is $\frac{3}{4}$. The numerator tells how many beats there are in each bar, and the denominator explains what is meant by a beat. In the case of $\frac{3}{4}$, there are three beats in every bar (nominator is 3) and each beat is defined as a quarter

Name	Note	Rest
Whole	𝅝	𝄻
Half	𝅗𝅥	𝄼
Quarter	♩	𝄽
Eighth	♪	𝄾
Sixteenth	𝅘𝅥𝅯	𝄿

Figure 2.4: Illustrating the duration of notes and rests used in sheet music.

note (denominator is 4). The two most common used time signatures are $\frac{3}{4}$ and $\frac{4}{4}$.

2.3 Scales and key of songs

A scale is a set of notes written in ascending order with a range from one note to the next similar note. The intervals between the notes defines the type of scale, and the two most common types are the major scale and the minor scale. A major scale is shown in Figure 2.5, which is the scale of C major. And Figure 2.6 shows the scale of A minor.

The melody of a song is often built around a scale, meaning that the most frequently used notes in the melody are the notes from the scale. The scale of which a melody is built around, is called the key of the song. In sheet music, the key of a song is represented by the key signature, which consists of the accidentals for the matching scale. Figure 2.7 gives an overview of the key signatures and their corresponding major and minor keys. As an example, looking at the excerpt from ”Silent Night” (Figure 2.2), we find that the song

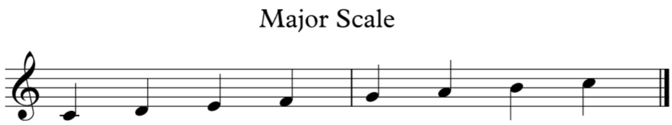



Figure 2.5: The scale of C major.

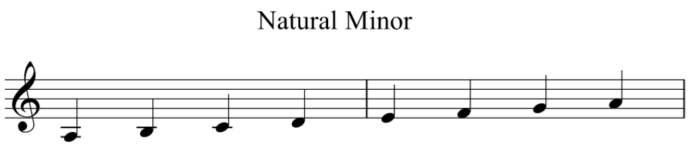



Figure 2.6: The scale of A minor.

is either in the key of F major or D minor.

The two keys that has the same accidentals, i.e. F major and D minor, are called relative keys. The two scales consist of the exact same notes, but the starting point of the scale differs. Because relative keys use the same notes, it can be hard to decide which key the song really is in. For example, how can we tell if Silent Night is in the key of F major or D minor? A way to get an impression of what the answer might be, is to consider the last note of the melody. It is common that this note is the same as the root of the key. The root of the key is the note giving name to the key, i.e D for D minor. Considering "Silent Night" (Figure 2.2), we see that the last note is F. Hence, the key of Silent Night is F major and not D minor.

All songs do not have the same key signature, but this can be obtained by transposing the songs. When transposing a song, all the notes are moved a fixed interval up or down in pitch. To make the analysis easier, the songs in this thesis are transposed so that they all have the same key signature. All major songs are transposed to C major and all minor songs to A minor. C major and A minor are relative keys and from Figure 2.7 we see that they have no accidentals. When considering a piano (Figure 2.1), the notes of these two

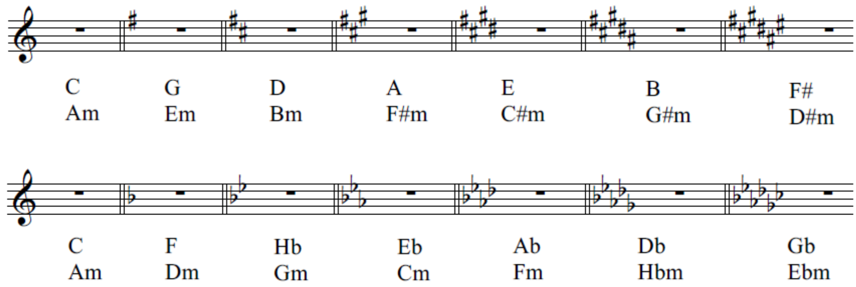


Figure 2.7: An overview of the key signatures and their corresponding keys.

keys are all the white tangents. The transposing of a song does not affect our problem of choosing chords, and hence this can be done to make our models less complex and the analysis easier. And after chords are predicted, the song and chords can be transposed back to the original key.

Figure 2.8 shows a transposed version of Silent Night to the key of C major. Comparing the transposed version to the original version in Figure 2.2, we see that the key signature now has changed. From containing a *b* in Figure 2.2 to now being empty. Also, all the notes are shifted to a higher pitch in the transposed version compared to the original version.

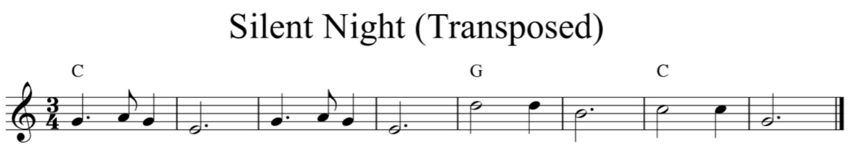



Figure 2.8: The excerpt from "Silent Night" transposed to the key of C major. All the notes are now shifted to a higher pitch.

2.4 Chords

Music as we are used to hear it, often consists of a melody and chords that accompany this melody. A chord is a combination of three or more notes, and are in sheet music denoted by a capital letter above the staff (see Figure 2.2). Typically, in modern music, the melody is performed by a singer or solo instrument and the other instruments (guitar, piano, bass etc.) use the chords to accompany the singer. There exists a lot of chords, but to simplify the problem, we choose to focus only on major and minor chords consisting of three notes.

The major chord is made up of the first, third and fifth step in the major scale. Likewise, is the minor chord made up of the first, third and fifth step of the minor chord.

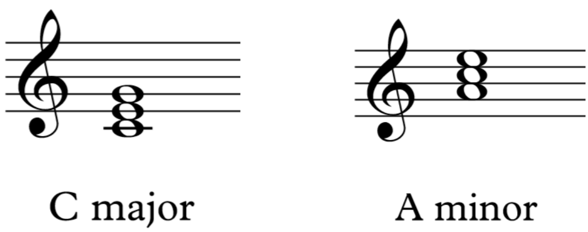



Figure 2.9: Showing the C major chord and the A minor chord using sheet music.

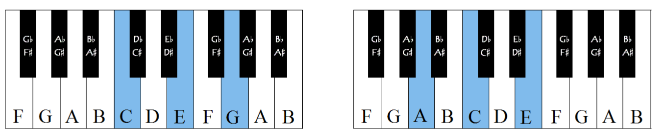


Figure 2.10: The chords C major and A minor illustrated using the piano. The notes appearing in the chord are marked with blue.

Figure 2.9 shows the chord C major and the chord A minor and Figure 2.10 shows the same two chords on the piano. To distinguish between major and minor chords in the sheet music notation, the subscript m is added to minor chords. Hence, denoting A minor as Am.

In Figure 2.10, the difference between the structures of the major and minor chord can easily be seen. Counting the tangents in the first interval of the major chord, between C and E, we find 3 free tangents. And there are 2 free tangents in the second interval, between E and G. For the minor chord, the relation between the notes are the opposite of the major chord. We find 2 free tangents between A and C, and three free tangents between C and E. The structure of a chord decides what type of chord it is. All major chord has the same structure as the one observed on the piano for C major, and likewise do all minor chords have the same structure as A minor.

2.5 Chord progressions

A melody has a specific key and uses the notes in the scale corresponding to this key. The same happens for the chords of the melody. A chord may be built upon any note of the musical scale; therefore, a seven-note scale allows seven triads, each degree of the scale becoming the root of its own chord (Chadwick, 1987). Figure 2.11 shows the triads that can be built upon the C major scale. Since the A minor scale consists of the same notes as the C major scale, we obtain the same chords for this scale. One of the chords in Figure 2.11 are a diminished chord, the chord B^o. This type of chord will not be considered in this thesis.

A commonly used notation for chords built on the scale are obtained by using Roman numerals, see Figure 2.11. A C major chord will be written I in the key of C, for example, but V in the key of F. Minor chords are represented by lower case Roman, so that D minor in the key of C would be written ii.

In Figure 2.11, there are three major chords. These chords are based on the first, fourth, and fifth note of the scale. They are often called the tonic, subdominant and dominant or using Roman numerals I-IV-V. In major key songs, these are the most frequently used chords. Together the three chords include every note of the scale, and can therefore harmonize all the scale notes.

When considering the minor scale, the chords based on the first, fourth and fifth note of the scale are all minor chords. They also include every note of the scale and are frequently used when harmonizing a minor key song. Sometimes, the

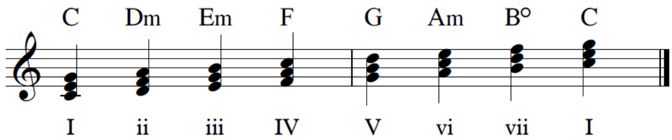


Figure 2.11: Showing the seven chords that can be built upon the scale of C major. The chords are numbered using Roman numerals.

fifth degree chord of a minor scale are exchanged with a major chord. In the case of A minor, we obtain the chord E major instead of E minor. This comes from a much related scale, called the harmonic minor. This scale is shown in Figure 2.12 and compared to the A minor scale, the seventh note G is raised half a step to G♯. This change of notes, results in an E major chord as the fifth degree chord.

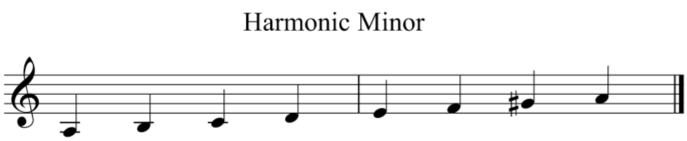


Figure 2.12: The harmonic A minor scale. The seventh note is raised half a step compared to the natural A minor.

A series of chords is called a chord progression. Often the same series are used in many types of music. Melodies can often be divided into sections of usually 8 or 16 measures, and it is quite common to repeat chord progressions, perhaps with small variations, in each section.

The simplest, and very much used, chord progression, is the series V-I. The fifth degree chord is in this progression said to lead to the root chord. This progression occurs in many major key songs. And there are even whole songs built entirely by the repetition of the two chords. The folk song "Polly Wolly Doodle" and the song "Achy Breaky Heart" are two examples of this. The corresponding progression for minor key songs are v-i or for harmonic minor V-i. Other examples of known chord progressions are I-V-vi-VI, which is widely used in for example pop music, and ii-V-I which is a well-known jazz chord progression.

2.6 Music genres

Musical pieces are often classified into genres. Some examples are pop, jazz, rock and classical music. The chords assigned to a melody, depend much on the genre of the musical piece. For instance, many pop songs are known for using only four chords, so-called four chord songs. And in the jazz genre, the chords often consist of more than three notes (adding the seventh, ninth or thirteenth step of the scale). There are often some characteristic chord progressions associated with specific genres. The blues genre is a good example of this, with its famous twelve-bare blues scheme. Thus, when making a model for chord prediction, it seems reasonable that the genre of the song must be taken into account.

Chapter 3

Hidden Markov Model

The models used for predicting chords in this thesis are variations of hidden Markov models (HMMs). In this chapter, we present some theory behind HMMs and we describe the two algorithms forward-backward and Viterbi algorithm.

3.1 Markov Chains

We define a random variable x that can take discrete values from a state space Ω_x of size N, $x \in \Omega_x = \{1, \ldots, N\}$. So, x can take one of N possible classes. The series $\boldsymbol{x} = (x_1, x_2, \ldots, x_T)$ over the time steps $t = 1, \ldots, T$ and for $x_t \in \Omega_x$, is a stochastic process. If the series follows a Markov property, then it is a Markov chain. The series is a kth order Markov chain if the conditional probability of the state at time step t given all the previous states, depends only on the last k states

$$p(x_t|x_{t-1}, \ldots, x_1) = p(x_t|x_{t-1}, \ldots, x_{t-k}).$$

Hence, a first order Markov chain has the property

$$p(x_t|x_{t-1}, \ldots, x_1) = p(x_t|x_{t-1}). \tag{3.1}$$

Figure 3.1 shows a graph representation of a first order Markov chain. The arrows represent the dependencies and illustrate the first order Markov property, since a node only depends on the previous node.

Figure 3.1: Illustration of a first order Markov chain. The arrows represent the dependencies between the nodes.

For a stationary Markov chain, the probability of going from one state i to another state j is the same for every time step of the series.

$$p(x_t = j|x_{t-1} = i) = p(x_{t+s} = j|x_{t+s-1} = i),$$

for any integers $s \geq 1$ and $t \geq 1$. This probability is called the transition probability and we denote it P_{ij}.

$$P_{ij} = p(x_t = j|x_{t-1} = i),$$

for all $t = 2, \ldots T$.

The transition probabilities are often represented by a (N×N) transition matrix, $\boldsymbol{P}$

$$\boldsymbol{P} = \begin{pmatrix} P_{11} & P_{12} & \ldots & P_{1N} \\ P_{21} & P_{22} & \ldots & P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \ldots & P_{NN} \end{pmatrix} \tag{3.2}$$

Because the matrix $\boldsymbol{P}$ consists of the probabilities P_{ij}, each row must sum to one

$$\sum_{j=1}^{N} P_{ij} = 1,$$

for all $i = 1, \ldots, N$. For a first order Markov chain, the probability mass function of the full stochastic process is given by

$$p(\boldsymbol{x}) = p(x_1) \cdot \prod_{t=2}^{T} p(x_t|x_{t-1}) \tag{3.3}$$

where $p(x_1)$ can be defined from an initial stationary distribution, and $p(x_t|x_{t-1})$ is defined in the transition matrix $\boldsymbol{P}$.

3.2 Hidden Markov Models

Now, suppose we have a first order stationary Markov chain $\boldsymbol{x} = (x_1, \ldots, x_T)$, for all $t = 1, \ldots, T$ and $x_t \in \Omega_x$. In each time step t, an effect can be observed, denoted y_t, and we get the series of observations $\boldsymbol{y} = (y_1, \ldots, y_T)$. The observations $\boldsymbol{y}$ are conditionally independent given $\boldsymbol{x}$ and they are also single site dependent on the elements of $\boldsymbol{x}$. In other words, the observation y_t depends only on the variable x_t. This leads to the likelihood distribution

$$p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{t=1}^{T} p(y_t|\boldsymbol{x}) = \prod_{t=1}^{T} p(y_t|x_t). \tag{3.4}$$

An observation $\boldsymbol{y}$ can be discrete or continuous. And in the next section, we outline the likelihood we use in the thesis. When only $\boldsymbol{y}$ is observed and the underlying variables $\boldsymbol{x}$ are unknown or "hidden", this defines a HMM. Figure 3.2 shows an illustration of a HMM. In the display, the top level (green) represents the observations $\boldsymbol{y}$ and the bottom level (blue) is a first order Markov chain similar to the one from Figure 3.1.

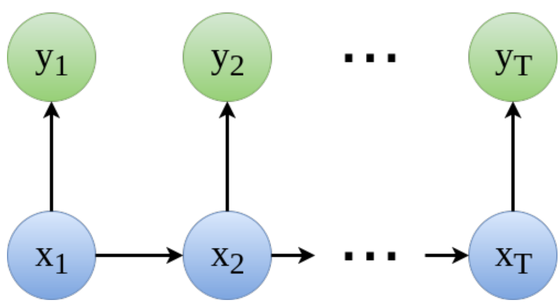


Figure 3.2: Illustration of a first order HMM. The top level (green) represents the observations $\boldsymbol{y}$ and the bottom level (blue) is a first order Markov chain $\boldsymbol{x}$.

When the observed values $\boldsymbol{y} = (y_1, \ldots, y_T)$ are known, and we want to find the probabilities of $\boldsymbol{x}$, we have an inverse problem. Bayesian inversion can be used to assess the hidden states (Kolbjørnsen, 2002). Bayes' rule states

$$p(\boldsymbol{x}|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x})}{p(\boldsymbol{y})}, \tag{3.5}$$

where $p(\boldsymbol{x})$ is the prior model given by equation (3.3) and $p(\boldsymbol{y}|\boldsymbol{x})$ is the likelihood model from equation (3.4). Now, when considering $p(\boldsymbol{y})$ as a constant term $C = p(\boldsymbol{y})^{-1}$ we obtain

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{x})p(\boldsymbol{x}), \tag{3.6}$$

which is the posterior model.

For a first order HMM, it is possible to factorize the posterior model and to obtain

$$p(\boldsymbol{x}|\boldsymbol{y}) \propto p(x_1) \prod_{t=2}^{T} p(y_t|x_t) \cdot p(x_t|x_{t-1}).$$

3.3 The likelihood model

We now define a likelihood model that is the same for every state t

$$p(y_t|x_t) = p(y_{t+s}|x_{t+s}),$$

for integers $s \geq 1$.

When the sample space of y is discrete with size M, meaning that y_t can take one of M different classes ($y_t \in 1, \ldots, M$), the likelihood model can be represented by a (M×N) matrix. This matrix is called the emission matrix and we denote it $\boldsymbol{E}$. Hence, given the probabilities

$$E_{ij} = p(y_t = j|x_t = i),$$

the emission matrix is given by

$$\boldsymbol{E} = \begin{pmatrix} E_{11} & E_{12} & \dots & E_{1M} \\ E_{21} & E_{22} & \dots & E_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ E_{N1} & E_{N2} & \dots & E_{NM} \end{pmatrix}. \tag{3.7}$$

3.4 The forward-backward algorithm

To assess the posterior probability $p(\boldsymbol{x}|\boldsymbol{y})$ from equation (3.6), the forward-backward algorithm (Baum et al., 1970; Scott, 2002; Lindberg, 2014) can be used. The algorithm can be divided into a forward step and a backward step, and the two parts are more thoroughly explained in the two following subsections.

3.4.1 Forward recursion

In the forward step, we iterate from $t = 1$ to $t = T$. We introduce the notation $\boldsymbol{y}_{1:t} = y_1, \dots y_t$. The forward recursion begins with finding the conditional probability given the observations in all the previous steps $p(x_t = j|\boldsymbol{y}_{1:t-1})$. Exploiting the Markov property of a HMM (3.1), we find that

$$p(x_t = j, x_{t-1} = i|\boldsymbol{y}_{1:t-1}) = p(x_t = j|x_{t-1} = i)p(x_{t-1} = i|\boldsymbol{y}_{1:t-1}), \tag{3.8}$$

and by summing over all the possible classes for i, we obtain

$$p(x_t = j|\boldsymbol{y}_{1:t-1}) = \sum_{i=i}^{N} p(x_t = j, x_{t-1}|\boldsymbol{y}_{1:t-1}).$$

Thereon we update the probabilities in (3.8), adding information from the observation y_t at the current time step t. This is done by using Bayes' rule (3.5) and exploiting the conditional independence property of the likelihood model (3.4). We obtain

$$p(x_t = i|\boldsymbol{y}_{1:t}) = \frac{p(y_t|x_t = i)p(x_t = i|\boldsymbol{y}_{1:t-1})}{p(y_t|\boldsymbol{y}_{1:t-1})},$$

where

$$p(y_t|\boldsymbol{y}_{1:t-1}) = \sum_{i=1}^{N} p(y_t|x_t = i)p(x_t = i|\boldsymbol{y}_{1:t-1}).$$

Initial values for the forward step are found by

$$p(x_1 = i|\boldsymbol{y}_{1:1}) = \frac{p(y_1|x_1 = i)p(x_1 = i)}{\sum_{i=1}^{N} p(y_1|x_1 = i) \times p(x_1 = i)}.$$

3.4.2 Backward recursion

The backward recursion starts where the forward recursion ends. Now we iterate backwards from $t = T$ to $t = 1$ and the goal is to find the backward probabilities $p(x_t = i|\boldsymbol{y})$. This is the probabilities of x being class i given all the observations $\boldsymbol{y}$. Making use of the Markov property (3.1), the backward probabilities can be found by

$$p(x_{t-1} = i, x_t = j|\boldsymbol{y}) = \frac{p(x_{t-1} = i, x_t = j|\boldsymbol{y}_{1:t})}{p(x_t = j|\boldsymbol{y}_{1:t})} p(x_t = j|\boldsymbol{y}),$$

and by summing over all the possible values of j we obtain the backward probability

$$p(x_{t-1} = i|\boldsymbol{y}) = \sum_{j=1}^{N} p(x_{t-1} = i, x_t = j|\boldsymbol{y}).$$

Then we can update considering the previous step as well by

$$p(x_t = j|\boldsymbol{y}, x_{t-1} = i)) = \frac{p(x_{t-1} = i, x_t = j|\boldsymbol{y})}{p(x_{t-1}|\boldsymbol{y})}.$$

3.4.3 Pseudocode

Pseudocode for the forward-backward algorithm is presented in Algorithm 1. In the pseudocode we denote the forward probabilities by

$$\begin{aligned}\alpha_t(i|\boldsymbol{y}_{1:t}) &= p(x_t = i|\boldsymbol{y}_{1:t})\\ \alpha_t(i,j|\boldsymbol{y}_{1:t}) &= p(x_{t-1} = i, x_t = j|\boldsymbol{y}_{1:t})\end{aligned}$$

Similarly, the backward probabilities are denoted by

$$\begin{aligned}\beta_t(i,j|\boldsymbol{y}) &= p(x_{t-1} = i, x_t = j|\boldsymbol{y})\\ \beta_t(i|\boldsymbol{y}) &= p(x_t = i|\boldsymbol{y})\\ \beta_t(j|i,\boldsymbol{y}) &= p(x_t = j|x_{t-1} = i, \boldsymbol{y}),\end{aligned}$$

We also introduce the following notation for the likelihood model

$$l_t(i) = p(y_t|x_t = i).$$

This is the likelihood of the data at time step t, when the state x_t is in class i.

When all the backward probabilities are obtained from the forward-backward algorithm, the full joint posterior can be assessed by

$$p(\boldsymbol{x}|\boldsymbol{y}) = \beta_1(x_1|\boldsymbol{y}) \times \prod_{t=2}^{T} \beta_t(x_t|x_{t-1}, \boldsymbol{y}).$$

Algorithm 1 The forward-backward algorithm

Forward step:

Initialize:

$\mathrm{C}_1 = \left[\sum_{i=1}^{N} l_1(i) \times p(x_1 = i)\right]^{-1}$

$\alpha_1(i|\boldsymbol{y}_{1:1}) = C_1 \times l_1(i) \times p(x_1 = i), \quad i = 1, \ldots, N$

for t=2, ..., T **do**

$C_t = \left[\sum_{i=1}^{N} \sum_{j=1}^{N} l_t(j) \times P_{ij} \times \alpha_{t-1}(i|\boldsymbol{y}_{1:(t-1)})\right]^{-1}$

$\alpha_t(i,j|\boldsymbol{y}_{1:t}) = C_t \times l_t(j) \times P_{ij} \times \alpha_{t-1}(i|\boldsymbol{y}_{1:(t-1)}), \quad i,j = 1, \ldots, N$

$\alpha_t(j|\boldsymbol{y}_{1:t}) = \sum_{i=1}^{N} \alpha_t(i,j|\boldsymbol{y}_{1:t}), \quad j = 1, \ldots, N$

end for

Finding the normalizing constant:

$C = \frac{1}{p(\boldsymbol{y})} = \prod_{t=1}^{T} C_t$

Backward step:

Initialize:

$\beta_T(j|\boldsymbol{y}) = \alpha_T(j|\boldsymbol{y}_{1:T}), \quad j = 1, \ldots, N$

for t=T, ..., 2 **do**

$\beta_t(i,j|\boldsymbol{y}) = \frac{\alpha_t(i,j|\boldsymbol{y}_{1:t})}{\alpha_t(j|\boldsymbol{y}_{1:t})} \times \beta_t(j|\boldsymbol{y}), \quad i,j = 1, \ldots, N$

$\beta_{t-1}(i|\boldsymbol{y}) = \sum_{j=1}^{N} \beta_t(i,j|\boldsymbol{y}), \quad i = 1, \ldots, N$

$\beta_t(j|i,\boldsymbol{y}) = \frac{\beta_t(i,j|\boldsymbol{y})}{\beta_{t-1}(i|\boldsymbol{y})}, \quad i,j = 1, \ldots, N$

end for

3.5 The Viterbi Algorithm

The Viterbi algorithm (Viterbi, 1967; Forney, 1973) is an algorithm that finds the most likely sequence of the underlying states in a HMM. This sequence is called the maximum a posteriori prediction (MAP) and is the sequence with the highest posterior probability

$$\hat{\boldsymbol{x}}_{MAP} = \arg\max_{\boldsymbol{x}} \{p(\boldsymbol{x}|\boldsymbol{y})\}.$$

The algorithm uses the technique of dynamic programming by breaking the problem into a collection of simpler sub-problems. For each time step $t = 1, \ldots T$ and for each possible class $j = 1, \ldots N$, the algorithm finds the most probable path that $x_{1:t-1}$ that ends in the state $x_t = j$.

We let $\gamma_t(j)$ represent the probability of the most probable path $x_{1:t-1}$ ending in state j at time step t, i.e.

$$\gamma_t(j) = \max_{\boldsymbol{x}_{1:t-1}} \{p(\boldsymbol{x}_{1:t-1}, x_t = j|\boldsymbol{y})\}.$$

Because of the Markov property of a HMM, the probability can be computed using the probability from the previous time step $\gamma_{t-1}(i)$ and the backward probabilities obtained from the forward-backward algorithm (Algorithm 1). We find that

$$\gamma_t(j) = \max_{i} \{\gamma_{t-1} \times \beta_t(j|i, \boldsymbol{y})\}.$$

Now we define a pointer to keep track of the most probable path. We let $m_t(j)$ be the backtrace pointer from $x_t = j$ such that it points back to the previous state x_{t-1} that resulted in the most probable path to $x_t = j$. When reaching the last iteration $t = T$, the pointers lead us through the most probable path when starting at the most probable state j_{max} given by

$$j_{max} = \arg\max_{j} \gamma_T(j).$$

Pseudocode for the algorithm is shown below in Algorithm 2.

Algorithm 2 The Viterbi algorithm

Initialize:

$\gamma_1(i) = \beta_1(i|\boldsymbol{y}), \quad i = 1, \ldots, N$

$m_1(i) \to 0, \quad i = 1, \ldots, N$

for t=2,..., T **do**

for j=1,..., M **do**

$\gamma_{t,t-1}(i,j) = \gamma_{t-1}(i) \times \beta_t(j|i, \boldsymbol{y}), \quad i = 1, \ldots, N$

$\gamma_t(j) = \max_i \{\gamma_{t,t-1}(i,j)\}$

$m_t(j) \to \arg\max_i \{\gamma_{t,t-1}(i,j)\}$

end for

end for

$j_{\max} = \arg\max_j \{\gamma_T(j)\}$

return $\hat{\boldsymbol{x}}_{MAP}$ by following the pointers starting at $m_T(j_{max})$

3.6 Second-order HMM

When the underlying Markov chain in a HMM follows the second order Markov property

$$p(x_t|x_{t-1}, \ldots, x_1) = p(x_t|x_{t-1}, x_{t-2}),$$

we have a second order HMM. The state t is now dependent on the two previous states. Figure 3.3 shows a graph representing a second-order HMM. The dependencies of the observations $\boldsymbol{y}$ are the same as for a first-order HMM. They are conditionally independent given $\boldsymbol{x}$ and also single site dependent on the elements of $\boldsymbol{x}$. The arrows in the figure illustrates this.

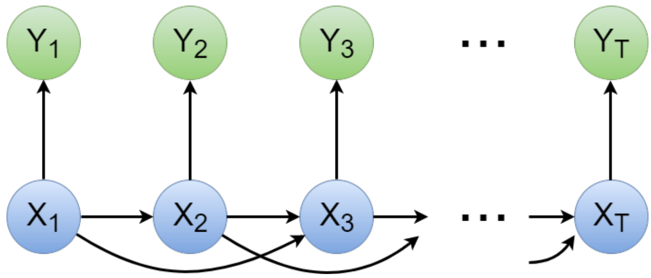


Figure 3.3: Illustration of a second order HMM. The arrows in the graph show the dependencies of the nodes. The top level (green) represents the observations $\boldsymbol{y}$ and the bottom level (blue) is a second order Markov chain.

A second-order HMM can be rewritten into a first-order HMM. In this way, the two algorithms presented earlier, the forward-backward algorithm (Algorithm 1) and the Viterbi algorithm (Algorithm 2), can be applied also to a second-order model. The rewriting is done by introducing an augmented state $\binom{x_t}{x_{t+1}}$ that contains both the state t and the next state $t+1$. This new augmented state is denoted $x_{t,t+1}$. This results in an increased sample size, we have $x_{t,t+1} \in \Omega_x \times \Omega_x$ which leads to $N_{aug} = N \times N$. Now, we have the dependency

$$p(x_{t,t+1}|x_{t-1,t}, \ldots, x_{1,2}) = p(x_{t,t+1}|x_{t-1,t}),$$

which is a first order Markov property. Figure 3.4 shows a diagram of HMM with augmented states.

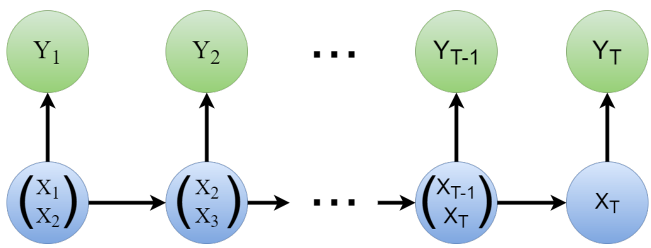


Figure 3.4: Illustration of a HMM with augmented states. The arrows in the graph show the dependencies of the nodes.

The likelihood model for this HMM depends only on the first state in the augmented state. Hence, we get

$$p(y_t|x_{t,t-1}) = p(y_t|x_t).$$

Chapter 4

The models for musical chord prediction

A melody with chords can be represented by a HMM (described in Chapter 3) by letting x_t represent the chord at time step t, and y_t represent the note appearing in the melody at time step t. In other words, the notes in the melody are considered to be observations and the chords are the hidden variables. In this chapter, we present the four models used for chord prediction.

In our models, we define the possible values of a note to be one of the twelve notes defined in Table 2.1. Each of the twelve notes can be the root of both a minor chord and a major chord. Thus, we get 24 possible chords. The possible notes and chords are shown in Table 4.1 together with labels. The four proposed models are Beat1, Measure1, Measure1m and Measure2. All the models are variations of HMMs, but the state definition and order of the models are varied. Each model is more thoroughly explained in the following sections.

4.1 Beat1

The Beat1 model is a first order model where the states are defined as single beats of the music. Figure 4.1 shows an illustration of this model. In the figure, the states x_t are represented in blue and contains the chord at time step t.

Note	Label	Chord	Label	Chord	Label
C	1	C	1	Cm	13
C$^\sharp$	2	C$^\sharp$	2	C$^\sharp$m	14
D	3	D	3	Dm	15
D$^\sharp$	4	D$^\sharp$	4	D$^\sharp$m	16
E	5	E	5	Em	17
F	6	F	6	Fm	18
F$^\sharp$	7	F$^\sharp$	7	F$^\sharp$m	19
G	8	G	8	Gm	20
G$^\sharp$	9	G$^\sharp$	9	G$^\sharp$m	21
A	10	A	10	Am	22
A$^\sharp$	11	A$^\sharp$	11	A$^\sharp$m	23
B	12	B	12	Bm	24

Table 4.1: The notes and chords considered in our models together with labels.

The length of the time step corresponds to the length of a beat in the song. Observations y_t are represented in green and consist of the first note appearing in the beat. The arrows represent the dependencies in the model. Using the notes and chords from Table 4.1 we obtain $y_t \in [1, \dots, 12]$ with $M = 12$ and $x_t \in [1, \dots, 24]$ with $N = 24$.

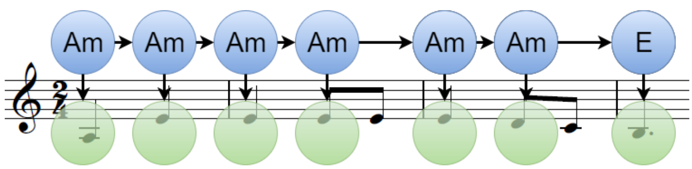


Figure 4.1: Illustration of Beat1. The states x_t are represented in blue and contains the chord at time step t. The length of the time step corresponds to the length of a beat in the musical piece. Observations y_t are represented in green and consists of the first note appearing on the beat. The arrows represent the dependencies in the model.

4.2 Measure1

The Measure1 model is also using a first order HMM to represent the song, but the states are defined differently than in Beat1. Now, a state consists of a whole measure and not just a single beat. Figure 4.2 illustrates the model. Again, the states x_t are represented in blue and contains the chord at step t. Observations y_t are shown in green and consists of all the notes in the measure.

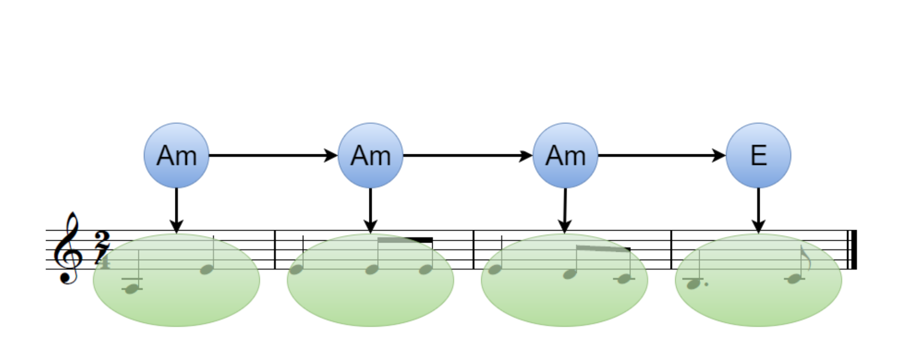


Figure 4.2: Illustration of model Measure1. The states x_t are represented in blue and contains the chord at time step t. The length of the time step corresponds to the length of a measure in the musical piece. Observations y_t are shown in green and consists of all the notes in a measure represented by an indicator vector. The arrows represent the dependencies in the model.

Since the observation y_t now may consist of multiple notes, an indicator vector is introduced to represent this in a suitable way. The vector indicates which notes are present in a measure. Each note is represented with an index, and the index of a note is given by the labels from 4.1. If a note is present in the measure, the corresponding index in the indicator vector will be set to 1, otherwise it is 0. This way of representing the measure does not take into account the number of times the note is present in the measure, nor the length of the notes. It only tells if a note is present or not.

An example is given below, where the four indicator vectors representing the observations y_t of the first four states $t = 1, \dots 4$ from Figure 4.2.

$$
\begin{aligned}
y_1 &= [0,0,0,0,1,0,0,0,0,1,0,0] \\
y_2 &= [0,0,0,0,1,0,0,0,0,0,0,0] \\
y_3 &= [1,0,1,0,1,0,0,0,0,0,0,0] \\
y_4 &= [1,0,0,0,0,0,0,0,0,0,0,1]
\end{aligned}
$$

Measure1 has the sample size $M = 2^{12} = 4096$ for the observations, and the sample size for the hidden states (chords) are the same as for Beat1, $N = 24$.

4.3 Measure1m

From music theory, we know that there is a difference between the chord distribution for a major key song and a minor key song. Thus, Measure1m build separate models for these two types. This is done by using the major songs as training data for one model, and the minor songs as training data for the second model. The models are built like the Measure1 model. They are first order HMMs using whole measures as state definition, where an indicator vector represents the observations.

Now, when predicting chords for an input melody, we first classify the song into either major or minor. This is done by a simple approach. We know that the final note in most melodies are the root note of the key of the song (see Section 2.3). Hence, after the melody is transposed to either C major or A minor, we use the last note of the melody to classify the song. If this note is an A, the song is classified as a minor key song, otherwise it is classified as a major key song.

Figure 4.3 shows an illustration of this model. The last measures of "Silent Night" is used as an example. First the song is classified as a major key song because the final note is a C, then the corresponding HMM is used for chord prediction.

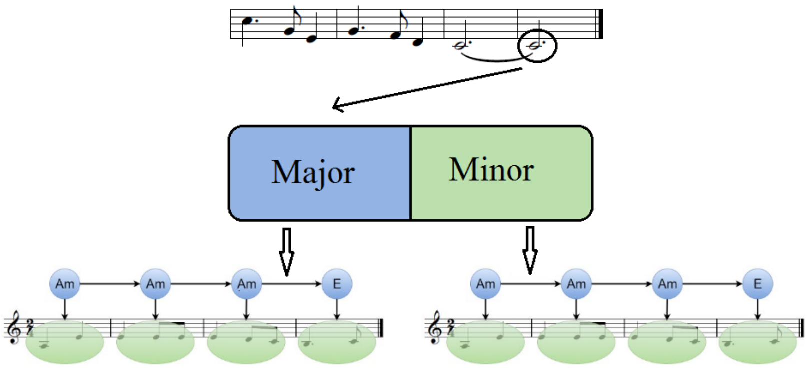


Figure 4.3: Illustration of model Measure1m. The four last measures from "Silent Night" is shown, and the last note is considered when classifying the song as a major song. Two separate models are built for major and minor. In both models, the states x_t are represented in blue and contains the chord at time step t. The length of the time step corresponds to the length of a measure in the musical piece. Observations y_t are shown in green and consists of all the notes in a measure represented by an indicator vector. The arrows represent the dependencies in the model.

4.4 Measure2

The last proposed model is Measure2, which is a second order model. The transition probabilities in the underlying Markov chain now depends on the 2 previous states as described in section 3.6. The model defines a state as a whole measure as for Measure1 and Measure1m, and it uses indicator vectors to represent the observations. An illustration of the model is shown in Figure 4.4.

As shown in section 3.6, this model can be rewritten to a first order model by introducing an augmented state $\binom{x_t}{x_{t+1}}$. This results in a sample sizes $M = 4096$

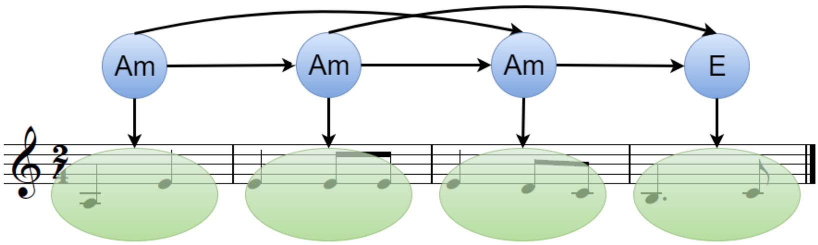


Figure 4.4: Illustration of model Measure2. The states x_t are represented in blue and contains the chord at time step t. The length of the time step corresponds to the length of a measure in the musical piece. Observations y_t are shown in green and consists of all the notes in a measure represented by an indicator vector. The arrows represent the dependencies in the model.

and $N = 24 \times 24 = 576$.

4.5 Overview of the models

An overview of the four models and some central properties is given in Table 4.2. The table shows the order of the models, the state definitions, if separate models is built for major and minor, and lastly the sample sizes N and M for respectively the hidden variables (chords) and the observations (notes).

Model	Order	State	Minor/Major	N	M
Beat1	1	Beat	No	24	12
Measure1	1	Measure	No	24	4096
Measure1m	1	Measure	Yes	24	4096
Measure2	2	Measure	No	576	4096

Table 4.2: Displays central properties of the four models proposed in this thesis.

Chapter 5

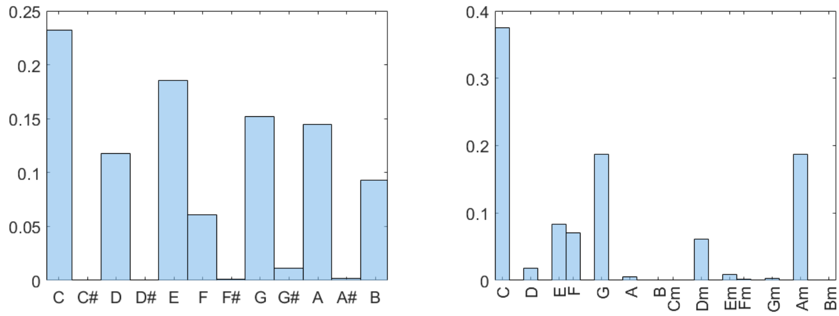
Training data and implementation

In this chapter the data used to train the models from Chapter 4, are presented. Also, some details related to the implementation of the models and the validation method are described.

5.1 Training data

To train the models, a data set of digital sheet music was used. The data set consists of in total 64 traditional children's song commonly sung in Norway. 22 of the songs are in a minor key and 42 of the songs are major key songs. A list of all the songs can be found in Appendix A. The data is collected from the two song books Sangskolen (Hukkelberg and Ekra, 1995) and Den store barnesangboka (Holen and Nordberg, 1985), and it is transcribed using the software Sibelius First to obtain the desired file type, MusicXML. The data consist of in total 1244 measures and 3297 notes.

The training data in this thesis is traditional children's songs. These are often built of simple chords and progressions, and this justifies the choice of only considering major and minor chords consisting of three notes. The distribution of the chords and the notes in the training are plotted in Figure 5.1. The



(a) Notes (b) Chords

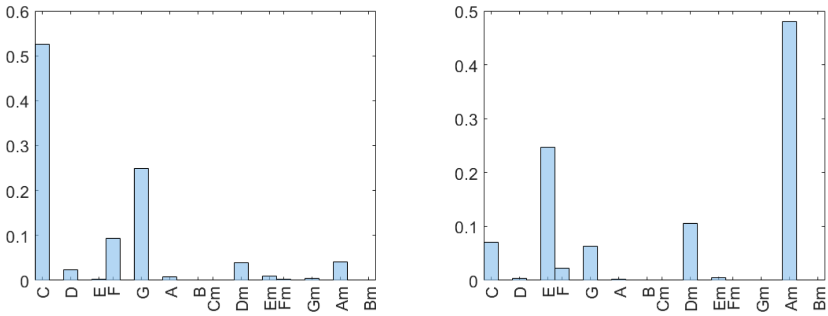
Figure 5.1: Histograms showing the distribution of the notes and the chords in the training data.

distributions represent the data after the songs are transposed to the key of C major or A minor.

Figure 5.1a shows that the most frequently occurring notes are the scale notes of C major or A minor, the notes without accidentals. This is expected, since a song is built around the scale corresponding to the key (see section 2.3). The three tones that are most represented in the training data are C, E and G. These three notes build the chord of C major (Figure 2.9). This is also as expected, since there are more major songs than minor songs in the data set.

The chord distribution plot in Figure 5.1b shows that the chord of C is the most common chord. This chord is used nearly twice as much as any of the other keys. Again, this is because of the many major key songs in the data set. In section 2.5 the three chords named the tonic, subdominant and dominant are mentioned. For C major these chords are C, F and G, and for the key of A minor the chords are Am,Dm and Em/E. From figure 5.1b we see that these chords are highly represented compared to the other chords, like the music theory indicates. The chord distribution plots also show that some of the notes and chords are never used. This will be further discussed in Chapter 6, when presenting the results.

Figure 5.2 shows the distribution of the chords after separating the major and minor key songs into two separate data sets. It can be seen that the chord distributions for the two types differ much. The tonic (C), subdominant (F) and dominant (G) are highly represented in the major chord distribution in



(a) Major songs (b) Minor songs

Figure 5.2: Histogram showing the distribution of the chords after separating the songs into major and minor key songs.

Figure 5.2a. The same can be seen for the minor chord distribution, the tonic (Am), subdominant (Dm) and dominant (E) are frequently used chords for the minor key songs. In both the major distribution and the minor distribution, we can see that the root chord, respectively C and A, has the highest representation.

5.2 Implementation

The article by (Nichols et al., 2009) proposes a pre-processing toolkit for the digital sheet music format MusicXML, and we use this toolkit in this thesis. The toolkit is written in MATLAB, and reads a data base of MusicXML files and creates a MATLAB struct with information about the songs. Amongst other things, this struct contains information about the notes of the melodies and the chords of the songs. Documentation can be found, and the script can be downloaded from the site `http://music.informatics.indiana.edu/code/musicxml`. The script transposes all the songs to either C major or A minor.

5.2.1 Estimation of transition and emission matrices

The training data was used to estimate the transition matrix (3.2) and the emission matrix (6.1). This was done by computing the relative frequency of the occurrence for each transition, given the training data. Hence, $\hat{P}_{ij}$ was found by counting the number of transitions from state i to j in the training data, and dividing by the total number of transitions from state i

$$\hat{P}_{ij} = \sum_t \frac{\#\{x_t = i, x_{t+1} = j\}}{\#\{x_t = i\}}.$$

The emission probabilities were found in a similar way, by

$$\hat{E}_{ij} = \sum_t \frac{\#\{x_t = i, y_t = k\}}{\#\{x_t = i\}}.$$

Since the different models from Chapter 4 have different state definitions, different transition and emission matrices, were obtained. After finding the matrices, the MATLAB function *hmmviterbi* was used to find the most likely chord progression given a melody input. This function is an implementation of the Viterbi algorithm explained in Algorithm 2.

5.2.2 Model validation

To analyse the performance of our models, the validation method leave-one-out cross-validation (LOOCV) (Wong, 2015) was used. The method leaves out one of the songs in the training data and build the models based on the remaining songs. Then the song that is left out is used as input data, and the chords are predicted for this song. This is done for every song in the training data, and the score is measured as the percentage of the correctly predicted chords. A similar and common way of validating the performance is by using k-fold cross-validation, but because of the small size of the training data, this method was not chosen.

Chapter 6

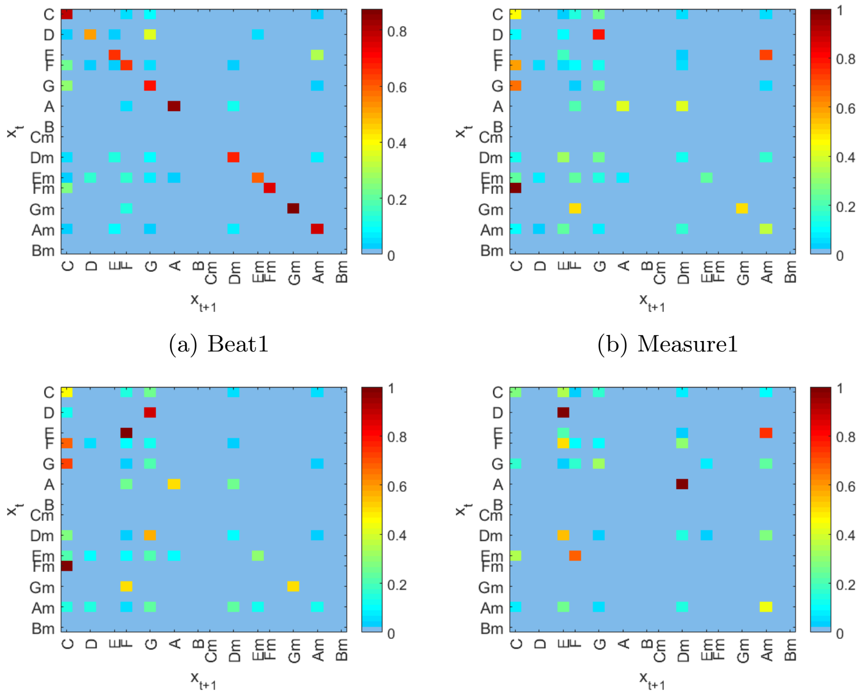
Results and discussion

The four models proposed in Chapter 4 were tested using the training data described in Chapter 5. This chapter presents the results after implementing and testing the models. First some general results are discussed, before we proceed to more specifically look at the results for three of the songs. The two major key songs "Silent Night" and "Fagert er landet" and one minor key song "Byssan lull".

6.1 Transmission matrices

Firstly, we look at the transition matrices as described in equation (3.2). The matrices for Beat1, Measure1 and two matrices for Measure1m (one for the major model and one for the minor model) are presented in Figure 6.1. The transition matrix for model Measure2 is not included, because of its large size. The color of the plot represents the value of the matrix, according to the color bars to the right of each plot.

From the figures, it can be seen that a lot of the transition probabilities are zero. According to Figure 5.1b in Chapter 5, a lot of the chords are not represented in our training data. The rows and columns for the chords not present, will all have zero probability. This is clear from the figure, and we can reduce the transition matrices by removing these rows and columns. These reduced transition matrices are displayed in Figure 6.2.



(a) Beat1

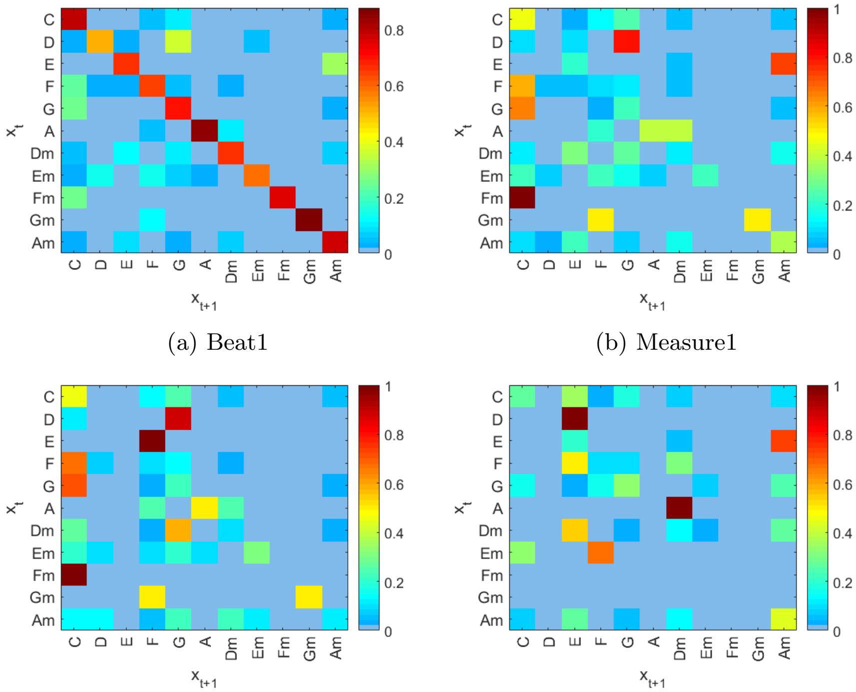
(b) Measure1

(c) Measure1 Major

(d) Measure1 Minor

Figure 6.1: Plot of the transition matrices for the different models. The color of the plot shows the value of each entry, according to the color bars to the right of each plot.

When considering the transition matrix for Beat1 (Figure 6.2a), we find high probabilities for two ascending chords to be the same. This is seen by the high probabilities occurring on the diagonal of the matrix. This model uses states consisting of only one beat, and since chords generally are held for whole measures consisting of multiple beats, there will be many transitions from one chord to the same chord. This property is a weakness for the Beat1 model, and leads to predictions with too few chord changes. This is clearly seen later, in the song examples.



(a) Beat1 (b) Measure1

(c) Measure1 Major (d) Measure1 Minor

Figure 6.2: Plot of the transition matrices of the four models after removing unused chords. The color of the plot shows the value of each entry, according to the color bars to the right of the plots.

The trend of high probabilities on the diagonal is less clear for the three other transition matrices (Figures 6.2b, 6.2c and 6.2d) for Measure1, Measure1 major and Measure1 minor. These models use states consisting of whole measures and this leads to less transitions between similar chords compared to Beat1.

By considering the two matrices built for Measure1m, the major matrix in Figure 6.2c and the minor matrix in Figure 6.2d, we can observe differences in the structure of major key songs and minor key songs. In the distribution plots from chapter 5, Figure 5.2a and Figure 5.2b, we observed that the most frequently used chords for major songs was C, F and G, and that the most used chord in minor songs was the chord Am, Dm and E. This is reflected in the transition matrices. In the major matrix we find high values in the columns representing C,F and G, and in the minor matrix we find high values in the columns representing Am, Dm and E. These high values mean that the probability of transitioning into these chords are high, and the probability for predicting one of these chords will be high.

The transition matrix for Measure1, Figure 6.2b, can be seen as a mix between the major transition matrix and the minor transition matrix. Because the total set of training data has a majority of major songs, the Measure1 matrix is most similar to the major transition matrix. Still, some central properties occurring in the minor transition matrix can be found in the Measure1 matrix. The transition $E \to A$ is an example of this. The probability of this transition is high for Measure1 and for Measure1m minor, but in the major matrix it is not. This transition can be recognized as the chord progression V-i, which was mentioned as a known chord progression for minor songs in section 2.5. Another chord progression that can be recognized is the II-V-I progression for major songs. In the major matrix, we find high probabilities of both II-V ($D \to G$) and V-I ($G \to C$).

6.2 Validation scores

The performance of the models was examined using LOOCV as described in section 5.2.2. Table 6.1 shows the obtained results.

Model	Validation score
Beat1	50.3%
Measure1	62.8%
Measure2	62.9%
Measure1m	70.1%

Table 6.1: Validation scores of the four models after using LOOCV.

An increase in performance is obtained by defining the states to be measures instead of beats. The validation score is increased by over 10% after this adjustment of the model. Measure1 and Measure2 perform quite similar. So, for our current training data, a second order model does not improve the result noteworthy. The hypothesis of testing a second order model, was that a lot of chord progressions are three or four measures long. Thus, thinking that a second order model could recognize these progressions. This does not seem to be the case for our test data. A possible reason, is that there is too little training data. Using a second order model leads to a large transition matrix. More training data could help to make a more even results for this matrix. Also, a second order model might be more useful when predicting chords for another music genre, which might have more complicated or longer chord progressions than children's songs do.

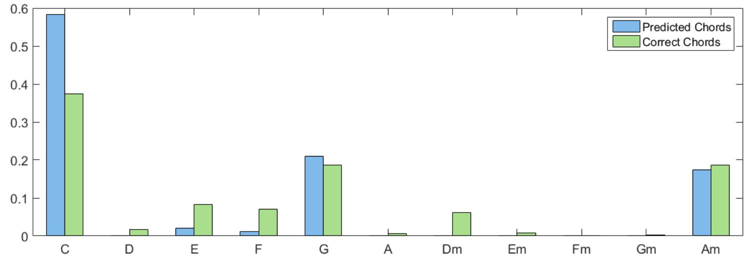
Building separate models for major and minor songs improves the predictions. The validation score increases from about 63% to 70%. Measure1m classifies all the input songs correctly into either minor or major. Because we have seen the difference in the chord usage for the two types, this is as expected.

A score of 70 percent for the best performing method is fairly well. A fact that must be precised when talking about the score of the models, is that there does not exist one correct solution to this prediction problem. More than one chord can be suitable for one measure of the melody. The same song can be harmonized in many different ways. This does much depend on genre and on the taste of the composer or song writer. Still, using the chords in the training data as the correct chords, does give a good picture of how well the models perform.

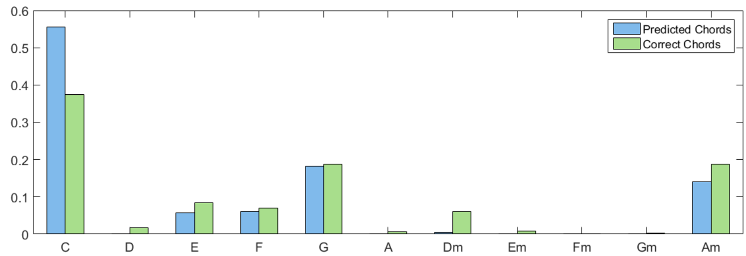
6.3 Distribution of predicted chords

In this section, we consider the distribution of the predicted chords. Histograms showing the distributions are shown in Figure 6.3. The chord distribution of the training data is plotted in the same figure for comparison. The predicted distribution is plotted in blue and the correct distribution is plotted in green.

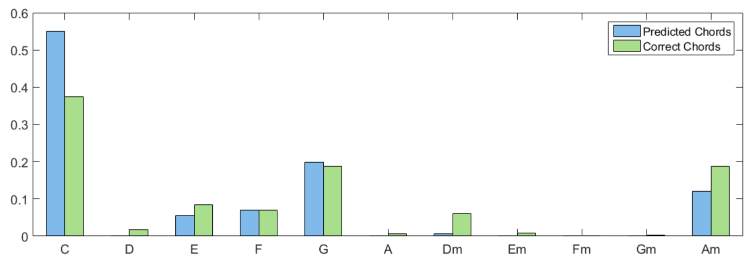
For model Beat1 in Figure 6.3a the variety of chords is very small. The chord C is overrepresented. This can be related to the transition matrix of the model (Figure 6.1a). When the probability of changing chord is small, and the chord C is a very likely chord to use, then there will be a lot of predictions choosing this chord.



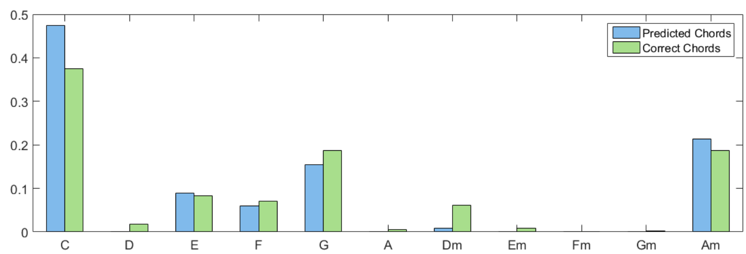
(a) Beat1



(b) Measure1



(c) Measure2



(d) Measure1m

Figure 6.3: Shows the distribution of the predicted chords (blue) together with the distribution of the correct chord distribution (green) for each of the four models.

Measure1 and Measure2 have very similar chord distributions. Comparing these two distributions to the Beat1 distribution, we see that the proportion of predicted C chords are lower. Still C is overrepresented compared to the correct distribution.

The distribution that is closest to the training data distribution, is the one for the Measure1m. This is as expected, since this also was the best performing model. In the distribution we find increased values for the chords E and Am compared to the distributions of the other models. Am and E are much used in minor key songs as we saw in the distribution plot in Figure 5.2b. Hence, the increase in prediction of these chords are because of the minor model that is built in the Measure1m model.

6.4 Discussion of prediction error

Now we proceed to discuss the prediction error by creating confusion matrices. These matrices can give us an impression of what errors our methods do. We let x_s be the correct chord corresponding to the prediction $\hat{x}_s$ at time s. The entries in the confusion matrix are then given by

$$\hat{C}_{ij} = \sum_{\boldsymbol{x}} \frac{\{x_s = i, \hat{x}_s = j\}}{\{x_s = i\}}.$$

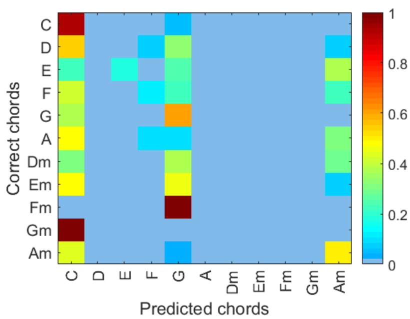
The value $\hat{C}_{ij}$ tells the relative number of predictions of the chord j given that the correct chord was i. The whole matrix is given by

$$\hat{\boldsymbol{C}} = \begin{pmatrix} \hat{C}_{11} & \hat{C}_{12} & \dots & \hat{C}_{1N} \\ \hat{C}_{21} & \hat{C}_{22} & \dots & \hat{C}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{C}_{N1} & \hat{C}_{N2} & \dots & \hat{C}_{NN} \end{pmatrix}. \tag{6.1}$$

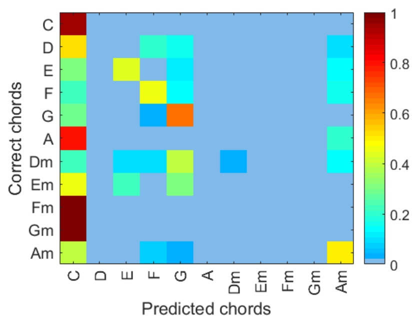
Figure 6.4 shows the confusion matrices for our four models. The values of the matrices are represented by the color defined in the colorbar. For a perfect prediction, the diagonal of this matrix would be one and the rest of the entries zero. This is not the case for our matrices.

These matrices are interesting because they tell what common mistakes our methods do. From the distribution plots of the predicted chords in Figure 6.3 we saw that the chord C was overrepresented in all the methods. This is clear from the confusion matrices as well. The column of C has high values, meaning that the chord C is often predicted even though it is not the correct chord.

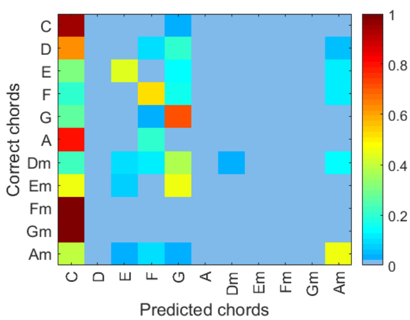
Considering the confusion matrix for Measure1m in Figure 6.4d we see that this method more often predicts the chords Am and E correctly compared to the other models. These chords are typical chords in a minor song, and hence we see that the Measure1m is a better model for minor songs.



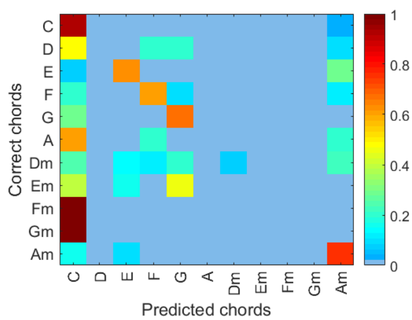
(a) Beat1



(b) Measure1



(c) Measure2



(d) Measure1m

Figure 6.4: Confusion matrices for the four models. The value is represented by the color defined in the colorbar to the right.

6.5 Silent Night

The first song we consider is "Silent Night". This is a popular Christmas carol composed by Franz Xaver Gruber. Figure 6.6 shows the prediction results obtained from our models and Figure 6.5 shows the sheet music for the song together with the predictions obtained from the Measure1m model.

From Figure 6.6 it can be seen that all the models perform fairly well for this song. Beat1 and Measure1 obtains the same result for this song, and also Measure 2 gets a similar result. All of these three models predicts the chord Am instead of the correct chord F. A reason for this mistake, might be that the chord Am are frequently used in minor key songs. Hence, a lot of the training data includes this chord, as seen in Figure 5.1). This is probably the reason why Measure1m is able to predict the F chord correctly. Measure1m classifies this song as a major key song, and in major key songs the chord F is more used than the chord Am.

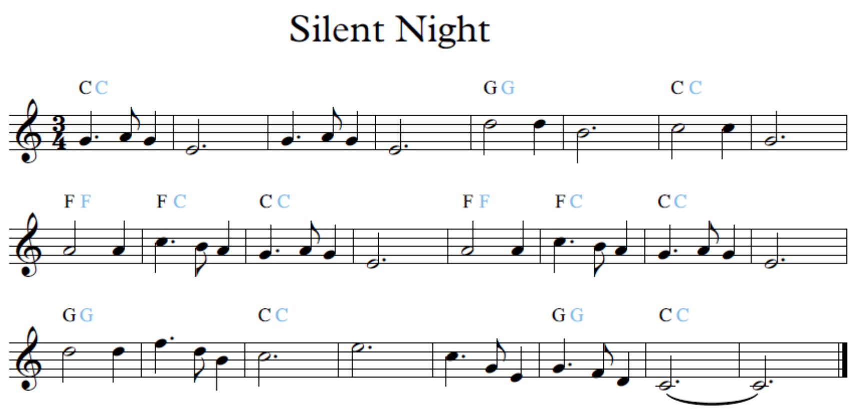
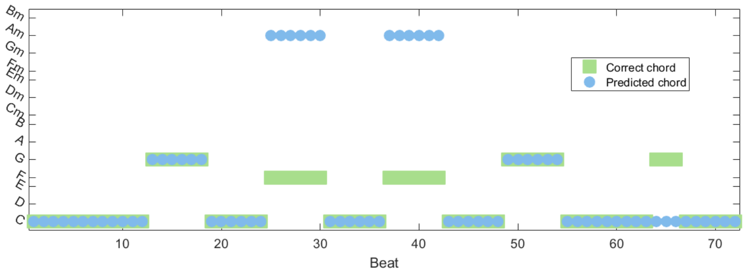
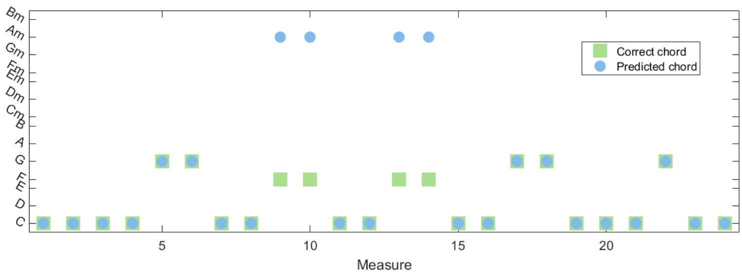



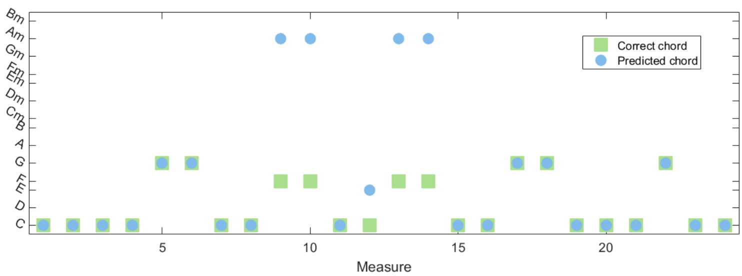
Figure 6.5: Sheet music representation of "Silent Night" showing the predicted chords obtained from the Measure1m model together with the correct chords. The correct chords are written in black and the predicted chords in blue.



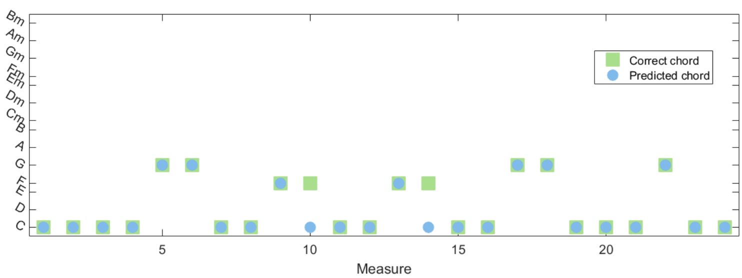
(a) Beat1



(b) Measure1



(c) Measure2



(d) Measure1m

Figure 6.6: Prediction results for "Silent Night". The predicted chords are plotted as blue circles and the correct chords as green squares.

6.6 Byssan Lull

Byssan Lull is used as a lullaby in Norway and is written by the Swedish composer Evert Taube. This is a minor key song. Figure 6.8 shows the prediction results obtained from our models and Figure 6.7 shows the sheet music for the song together with the predictions obtained from the Measure1m model.

This song illustrates the need of a separate minor key model. The three models Beat1, Measure1 and Measure2 performs very poorly compared to the Measure1m model. The Beat1 model in Figure 6.8a has only two correct predictions, the last predicted Am. The Beat1, Measure1 and Measure2 model predicts typical major chords like G major and C major for this song, whilst the Measure1m model classifies the song correctly as a minor key song and hence chooses more typical minor chords like Am and E.

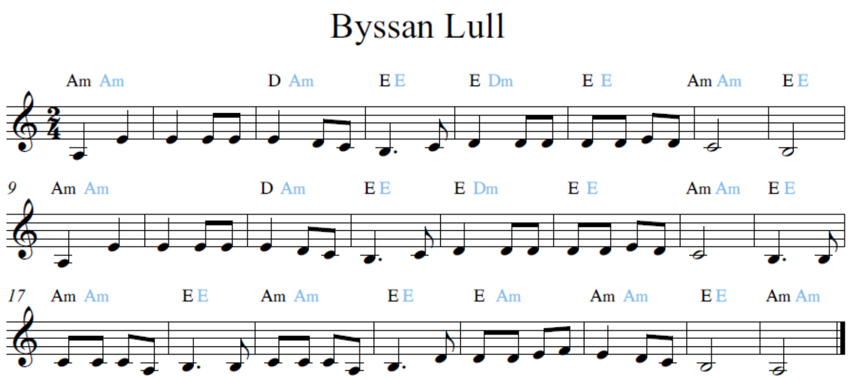
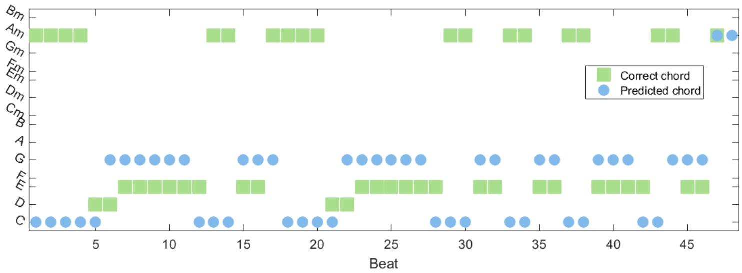
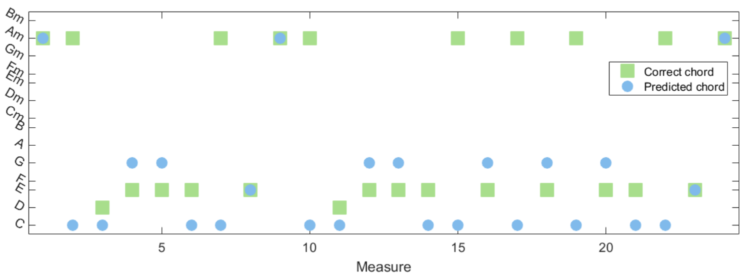


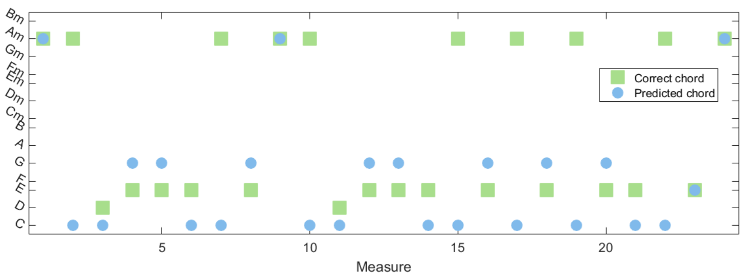
Figure 6.7: Sheet music representation of "Byssan lull" showing the predicted chords obtained from the Measure1m model together with the correct chords. The correct chords are written in black and the predicted chords in blue.



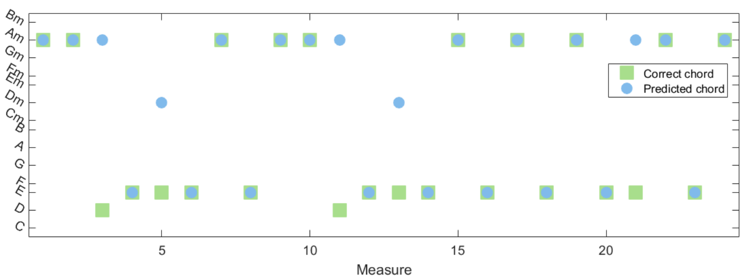
(a) Beat1



(b) Measure1



(c) Measure2



(d) Measure1m

Figure 6.8: Prediction results for "Byssan lull". The predicted chords are plotted as blue circles and the correct chords as green squares.

6.7 Fagert er landet

The last song studied, is the song "Fagert er Landet". This is a Norwegian hymn written by Anders Hovden. Figure 6.10 shows the prediction results obtained from our models and Figure 6.9 shows the sheet music for the song together with the predictions obtained from the Measure1m model.

None of the models performs well with this song. As previously mentioned, the Beat1 model has too many C major chord predictions and small variation in the chords. The three models Measure1, Measure2 and Measure1m predicts the exact same chords throughout the whole song. What is interesting about this hymn, is that it seems to change the mode (or key) of the song around measure 17. Up to this measure, the correct chords are typical major song chords like C, F and G.

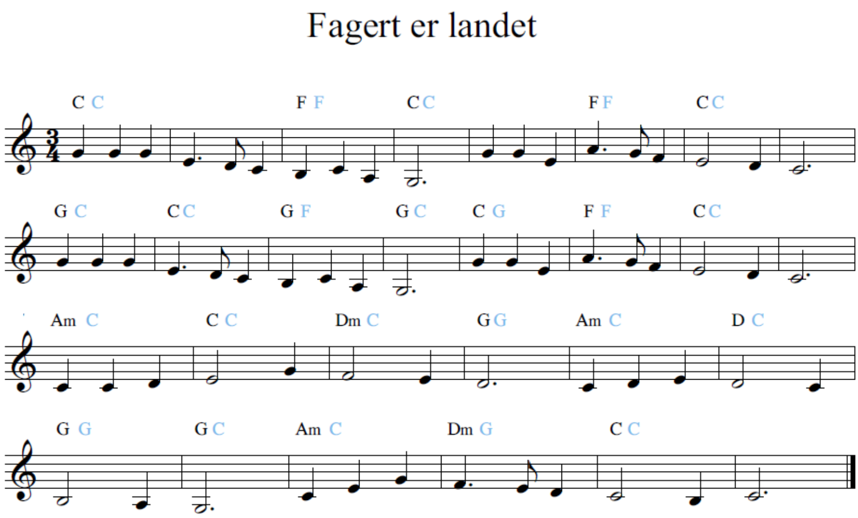
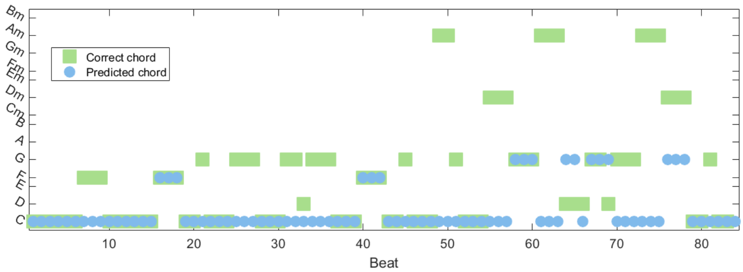
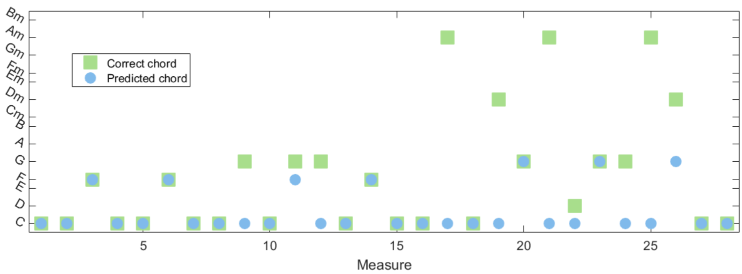



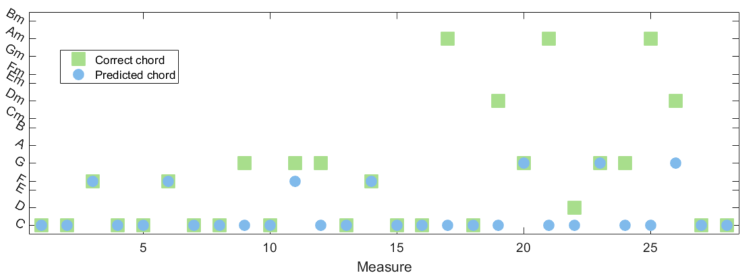
Figure 6.9: Sheet music representation of "Fagert er Landet" showing the predicted chords obtained from the Measure1m model together with the correct chords. The correct chords are written in black and the predicted chords in blue.



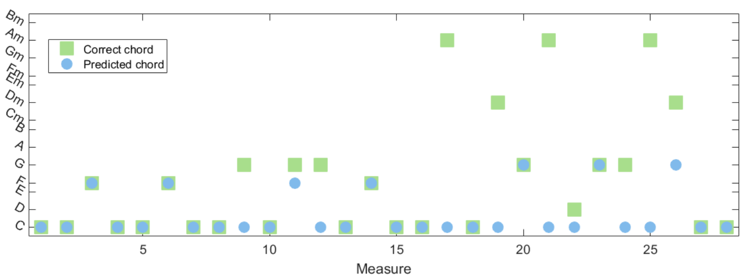
(a) Beat1



(b) Measure1



(c) Measure2



(d) Measure1m

Figure 6.10: Prediction results for "Fagert er landet". The predicted chords are plotted as blue circles and the correct chords as green squares.

But in measure 17, the song uses more of typical minor chords like Am and Dm. This is where the Measure-models fail to predict the chords. To correctly predict the chords for this song, one would need a model that recognizes this change of mode.

This song shows the reluctance of chord change in the Beat1 model. From Figure 6.10a we can see that nearly all the predicted chords are C, and that there are very few changes of chords compared to the correct chord progression and also compared to the predictions obtained from the other models.

Chapter 7

Closing remarks

This chapter is the closing chapter of the thesis. First a summary is given, before recommendations and further work are discussed.

7.1 Summary

In this thesis four models for chord prediction given a digital sheet music melody input are proposed. All of the models are HMMs, but the definition of the state and the order of the models are varied. One of the methods also builds two separate models, one used for minor key songs and one for major key songs.

Improvement of the models are obtained when using measures as state definition instead of single beats. The model Beat1 using the single beat state definition leads to too few chord changes because of the high transition probabilities of two adjacent chords being equal. Using measures as state definition helps with this problem. The second order model Measure2 performs very similar to the first order model Measure1 and does not lead to a noteworthy improvement of the predictions. The model is more complex and has a much larger sample size than the other models.

The best performing model Measure1m obtains a result of 70 % correctly predicted chords. This model classifies the input data as either minor or major and then uses a corresponding model for prediction. From plots of chord distributions, we found that the chord progression used for the minor key songs and major key songs are quite different. Because our training data consists of more major key songs than minor key songs, the minor key songs seemed to be neglected by the general models, and hence often wrongly predicted. Measure1m, on the other hand, predicted the minor key songs much better.

The models implemented in this thesis can be of good assistance when choosing chords for a melody, but they are not optimal. If used in addition to human interaction, the result can be quite good. Since the problem of choosing chords does not have one true solution, this can be considered as a good result. The chords chosen are often suitable, although they are not the preferred chords for a musician.

The structure of the chords progression for major key songs and minor key songs are different. Hence, a recommendation for further work is to use models that exploits these differences. Considering measures instead og single beats is also very effective.

Some proposals for further work, are to use different training data. Especially, would it be interesting to see how well the models perform with music from other musical genres. Increasing the size of the data set by adding more songs, would also be interesting. Perhaps would the second order model perform better by doing this. Also, considering the song example "Fagert er landet", we found that the models performed poorly when the melody changed mode. Creating a model that could recognize such changes in the melody would probably better the predictions. A lost proposal for further, is to change the representation of the measure notes. In this thesis, we used an indicator vector to represent the notes, but this does not give any information about the length of the note in the measure. A representation containing this information, is worth testing.

References

Allan, M. and Williams, C. (2005). Harmonising chorales by probabilistic inference. *Advances in Neural Information Processing Systems*, 17:25–32.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(3):164–171.

Chadwick, G. W. (1987). *Harmony: A course of study.* Boston: The B.F. Wood Music Co.

Chen, Z., Qi, J., and Zhou, Y. (2015). Machine learning in automatic music chords generation.

Cunha, U. S. and Famalho, G. (1999). An intelligent hybrid model for chord prediction. *Organised Sound*, 4(2).

Forney, G. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Ghahramani, Z. (2001). An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(1):9–42.

Holen, A. and Nordberg, H. (1985). *Den store barnesangboka.* Aschehoug.

Hukkelberg, L. and Ekra, S. G. (1995). *Sangskolen 1.* Harald Lyche Co Musikkforlag.

Ibe, O. (2013). *Markov Processes for Stochastic Modelings.* Elsevier Science.

Kolbjørnsen, O. (2002). Fundamentals of inverse problems. *Statistics*, 6(02).

Krogh, A., Brown, M., Mian, I., Sjolander, K., and Haussler, D. (1994). Hidden markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531.

Levinson, S. (1986). Continously variable duration hidden markov models for automatic speech recognition. *Computer Speech and Language*, 1:29–45.

Lindberg, D. V. (2014). Inference and categorical bayesian inversion of convolved hidden markov models applied to geophysical observations. *PhD in Mathematical Sciences, NTNU.*

MacDonald, I. and Zucchini, W. (1997). *Hidden markov and Other Models for Discrete-valued Time Series.* Chapman & Hall.

Nichols, E., Morris, D., Basu, S., and Raphael, C. (2009). Relationships between lyrics and melody in popular music. *Conference: Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International.*

Picone, J. (1990). Continuous speech recognition using hidden markov models. *IEEE ASSP Magazine*, 7(3):26–41.

Rabiner, L. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *IEEE Acoustics, Speech & Signal Processing Magazine*, 3:4–16.

Scott, S. L. (2002). Bayesian methods for hidden markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, 97(457):337–351.

Suzuki, S. and Kitahara, T. (2014). Four-part harmonization using bayesian networks: Pros and cons of introducing chord nodes. *Journal of New Music Research*, 43(3):331–353.

Thompson, E. (1983). Optimal sampling for pedigree analysis: parameter estimation and genotypic uncertainty. *Theoretical Population Biology*, 24:39–58.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 10(6):402–415.

Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition*, 48(9):2839–2846.

Appendix A

List of the songs used as training data

This appendix presents a list of the songs used as training data. The songs are a collection of traditional children's song much used in Norway. The data is collected from the two song books Sangskolen (Hukkelberg and Ekra, 1995) and Den store barnesangboka. (Holen and Nordberg, 1985). It consits of in total 64, where 22 of the songs is in a minor key and 42 of the songs are major key songs.

17.mai sang for de små
Alle fugler små de er
Amazing Grace
Auld lang syne
Ballade om en munk
Bind deg ein blomekrans
Bjørka
Blåmann Blåmann bukken min
Blinke blinke stjernelill
Byssan lull
Deck the halls
Den første løvetann
Den fyrste song
Det bor en baker
Du lirande lerke
Edelweiss
En bussjåfør
Er det ikke rart
Fader Jakob
Fager kveldsol smiler
Fagert er landet
Fløy en liten blåfugl
Fola fola blakken
God morgen alle sammen
Greensleeves
Gubben Noa
Happy Birthday
Hei Diddelumkum
Hevenu Shalom
Hu hei! Kor er det vel friskt og lett
Hurra for deg
Hvem kan seile
If you're happy and you know it
Ingerid Sletten
Jeg gikk en tur på stien
Julekveldsvise
Kom, skal vi klippe sauen
Kona og folungen
Lisa gikk til skolen
Måne og sol
Maria går blant tornekratt
Mellom bakkar og berg
My bonnie
No livnar det i lundar
O jul med din glede
Og hver og en
Oh when the saints
Old McDonald
Om kvelden
Pål sine høner
Per spelmann
Riv ned gjerdene
Såg du noko
Scarborough fair
Se min kjole
Ser du sola du Ola
Silent Night
Sov no liti tuve
Ta den ring
Trollmors vuggesang
Vi har ei tulle
Vise for gærne jinter
Vuggevise
When you're smiling