Thesis

A Thesis
Presented to
The Division of Mathematics and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Emily Palmer

May 2018

Approved for the Division
(Mathematics)

Andrew Bray

Acknowledgements

I want to thank a few people.

Table of Contents

List of Tables

List of Figures

Abstract

The preface pretty much says it all.

Second paragraph of abstract starts here.

Dedication

You can have a dedication here if you wish.

Chapter 1

1.1 Introduction:

In the digital age data sets are everywhere. Billions of new data are generated daily, in banking, social media, and in other scientific studies. These data can be in numerous forms. With the availability of the internet and social media data, text classification has become an interesting form of data. While text classification problems are very frequent, similar methods that instead classify music have not been explored as much.

Thinking of text or music as data has its own challenges. On its own text as data does not contain anything immediately analyzable. The same goes for music as data. First, features must be created that can be extracted for every text. These can include features such as word frequency, length etc. Analysis of text using machine learning can find patterns, authors, or categories of text. Similar methods can in fact be used to classify music.

The essential part of either text or music classification is feature selection. Unlike in a data set of numerical or categorical values, text and music must first go through a processing stage where features of interest can be extracted before any models can be fit.

For music, the interest is in building a model that uses the extracted features would be able to correctly classify a likely composer for that piece of music. For musically trained humans, this might be an easy task. Some are able to either by listening to a recording, or looking at sheet music, automatically distinguish a piece composed by Bach or Mozart. This becomes more difficult when composers are contemporaries. Mozart and Saliery might be distinguishable to a scholar or music fan, but it might be harder. Harder still is when a piece of music has a disputed composer history. These examples exist throughout music history, most notably in the Renaissance. Music classification has attempted to assign composers to pieces to be disputed to be Josquin Des Prez.

In both text and music classification, we must create features that can be calculated that would give some signal to indicate some unconscious tendency of the composer that would make them distinguishable.

There have been numerous other applications of machine learning to music. Previous studies have used machine learning trained on a composer's music to have the computer learn the style and and compose a piece in a similar style. (Papadopoulos & Wiggins, n.d.)

1.2 Brief history of text classification

One of the earliest instances of text classification for determining authorship was on the Federalist papers.(Mosteller & Wallace, 1964). The famous Federalist Papers were written under the pen name 'Publius'. There are several disputed papers attributed to James Madison or Alexander Hamilton. The authors never admitted authorship, as some of the writings were contradictory to their later political platforms. (Adair, 1944) Historians have often examined the papers using styles of previously known writings of Madison and Hamilton. Their analysis is often partially based on the content of the letters, for example the existence of citing English history is a trait more common to Hamilton. (Ford & Bourne, 1897)

In contrast, using the frequency of words such as and 'by', 'from', and 'upon', Mosteller and Wallace trained the writings on a set of known writings by each author. These unconscious indicators were able to differentiate between the two writers, and when a model was trained, the model was able to identify the likely author of the disputed paper.

1.3 Feature selection

Text analysis, such as in the federalist papers, is read one word after another. Information in piece of music, however, is read in a variety of ways. It can be read left to right note by note, but it can also be read vertically as the harmony, or the notes played together. Also in a piece with several instruments, the above happens at the same time for each instrument. There are also aspects that take place over large sections, such as phrasing, or cadencial patterns. There are rules of counterpoint that are followed throughout the entire piece. Thus we need to find features that can be measured for each piece, or perhaps each measure or instrument, that can describe a certain piece of music. Then we must decide which features are those of rules and practices of classical music, and where the creativeness and individuality of a composer happens. As there are many different aspects of a piece, melodic changes, harmonies, etc, we can end up with many features to describe one piece. As in this instance when we only have ~150 total pieces, we can have that $p > n$. We thus need to figure a clever way to decrease our predictor space. This can happen in choosing what features we want to use to model, or using a dimension reduction technique such as PCA.

Most of the musical stylometry papers have focused on composers in the Renaissance, Baroque, and Classical eras. The Mendelssohns were composing in the Romantic period. This choice might be because composers in earlier eras followed rules of counterpoint more exactly, or perhaps had less "expressive" allowances for their composing, thus making features easier, although this is just speculation. There are also more pieces with doubtful authorship in those eras. In addition Computer Assisted Research in the Humanities (CCARH) has a large corpus of encoded music from these times. As will be described later, having music in a format immediately conducive to analysis is a huge time saver.

1.4 A Very Short Introduction to Music Theory

Western sheet music is presented on a five line staff. The vertical distance between notes (also known as an interval) depends on how many half steps occur between two notes. There are 12 half steps in the western scale. Melodic intervals are defined as the number of half steps between two adjacent notes. Harmonic intervals are defined as the number of half steps between notes played at the same time. Cadences are a type of chord progression, usually occurring at the end of a phrase and especially at the end of a piece. Musical notes are in the set note $\in \{A, B, C, D, E, F, G\}$ The value of a note can be changed up or down a half step, by adding a sharp or flat. There are 12 unique note value per octave. The key signature of the piece indicates how many sharps or flats are normal for the piece. A pieces key can be major or minor. The possible keys are in the set key $\in$. Intervals and chords can be either dissonant or consonant. Consonant intervals sound nice to our ears, whereas dissonant intervals add a sense of tension or unease, which is used to shape the feel of the music. In addition chords and intervals can be minor or major. Minor intervals feel “sad”, whereas major intervals feel “happy”.

For classical music there are some “rules” that most composers follow to some extent or another. These were followed more strictly in the Baroque era than the Romantic era. These rules are known as counterpoint. First species counter point is a type of counterpoint. One rule is that there are no parallel perfect consonances. Perfect consonances are octaves, fourths and fifths. Parrallel indicates that they occur one after another. Another rule is that from one perfect consonance to another, the must proceed in contrary or oblique motion. This means that if a fourth interval wants to move to a fifth, that the direction of increase must be opposite.

1.5 Musical features

Features calculated from music often closely follow music theory. In music, in addition to deciding the features, there is also the decision of the scope at where those features take place. They can be features for a given instrument, the entire piece, or each measure. Also windowing techniques can be used where a “window” is created over a given number of bars or notes, and shifts across the whole piece For each window, a feature is recorded. These can be overlapping windows by creating an “offset” of a number of beats or notes. This produces more data, as instead of one feature for each piece, there is a feature for each window.

Musical features are often thought about as either high level, or low level. While the exact definition of each often varies, low level is often understood to be features such as note frequency, etc. High level features are more about a broader sense of the piece, including chord progressions, etc. High level features are often what music experts use in their analysis, whereas low-level features are more easily done with computer analysis.

Especially in research regarding gene expression and text categorization, data sets have enormous numbers of variables. We use “feature” and “variable” interchangeably, with the exception when features are created from variables, and the distinction will be made in that case. (Guyon & Elisseeff, 2003). There are several variable selection algorithms that select the "important" variables. If we included every variable that we extract from the piece, our model would very likely be over fit.

The start of feature selection is domain knowledge. Thanks to John Cox in the music department for suggesting a list of valuable features. These will be described in Chapter 2.

1.5.1 Background on Variable and Feature Selection

Often, especially in musical research, before analysis is done, numerous features are extracted from the music without knowing a prior i which ones will be helpful in identifying a composers' style. Thus we have to choose which features we want to use in our model. There are many ways to do this. There are existing variable selection algorithms that can help with this process.

Several variable selection algorithms include variable ranking. Variable ranking uses a score function to assign a score to each possible variable. It is a computationally efficient method and is robust against over fitting as it introduces bias but may result in less variance. It is tempting to only include variables that have a high score. However, this possibly leads to redundancy. In addition, variables that are not important by themselves can have a significant performance improvement when considered with other variables. Popular variable ranking methods for classification are single variable classifiers and information theoretic ranking criteria.

Single variable classifiers rank the variable according to their individual predictive power. The predictive power can be measured in terms or error rate, or using the false positive or false negative rate (fpr, fnr). This classifier cannot distinguish variables that perfectly seperate the data.

After knowing the ranking of a variable we then select which variables will be useful for our model. This is known as variable subset selection. The three most common types of variable subset selection are wrappers, filters, and embedded methods. Filters do not involve any machine learning to create the criteria for variable subset selection. Wrappers on the other hand use the "performance of a learning machine trained using a given feature subset." Embedded methods perform variable selection in the process of training and are usually specific to given learning machines

All possible subsets of variables is $2^p - 1$, which for large p is often computationally impossible. Strategies like best-first, branch and bound, simulated annealing, and genetic algorithms can help with the computational difficulties.

Dimensional reduction is also commonly used to combine enough of the information given in the features in a smaller dimensional space. This results in feature creation; using the recorded variables to create new features to fit the model on. Dimensional reduction techniques include include clustering and PCA/SVD. The two goals of these feature creations are that we can achieve a good reconstruction of the data. The second is that we can be most efficient in making our predictors. The first is an unsupervised problem. The second is supervised.

1.6 Previous research.

Research on musical stylometry focuses on two areas, data in the form of audio, and data in the form of sheet music. Our analysis uses data in the form of sheet music. To predict a composer, a training data set of pieces of known composer is needed. Then that model can be fit to a testing data set to predict composer. If that shows good predictions, that model can be used on the pieces of unknown authorship. Musical stylometry can be used disputed authorship, but also to detect distinguishing musical styles of composer, even if there are no disputed pieces.

To my knowledge there are currently four studies that have looked at the question of classifying composers by extracting features from sheet music. Several previous papers have focused on Josquin des Prez. This is likely due to the fact that there is a large training and testing data set available in easily analyzable format provided by the Josquin Research Project ("The josquin research project," n.d.). In addition there are a number of pieces of disputed authorship that have been attributed to him.

Work by Brinkman et al. (Brinkman, Shanahan, & Sapp, n.d.) use machine learning approaches to evaluate attribution of compositions by Josquin des Prez. They first compared music of Josquin and JS Bach's four part chorals. Many listeners could easily distinguish these two composers as they were separated by two centuries. They also compared Josquin to composers closer in time and style: Ockeghem and Du Fay, who are only one or two generations separated. These composers could likely be differentiated by experienced listeners of Renaissance music. Finally they compared Josquin's contemporaries, de Orto, and La Rue.

Work by Speiser and Gupta (Speiser & Gupta, n.d.) also analyzed Josquin and his contemporaries to attempt to classify unknown works. They had 130 works by Josquin, 93 of Ockeghem, 183 of la Rue, and 287 works possibly attributed to Josquin.

Work by Backer et. al.(2005) looked at differences in style between J.S. Bach, Telemann, and Handel,Haydn and Mozart and then examined a disputed piece: BWV 534 which is believed to be composed by J.S. Bach, J.L. Krebs, or W.F. Bach (J.S. Bach's son). (Backer & Kranenburg, 2005) The above techniques were used to analyze the music of Bach, Handel, Telemann, Mozart and Haydn and compare J.S. Bach, W.F. Bach and J.L Krebs in an attempt to classify BWV 534.(Backer & Kranenburg, 2005)

Means et. al. (Mearns, Tidhar, & Dixon, 2010) did an exploration of extracting high level musical features on J.S. Bach, Buxtehude, Ruggero, Vivaldi, Monteverdi, Corelli, and Frescobaldi. The idea was that they all adhered to the laws of counterpoint, and that they possibly still had distinct uses of the counterpoint. They used pieces across instrumental genres with the idea that stylistic counterpoint would remain constant across com positional type.

1.6.1 Previous choices of features in applications

Deciding on and extracting features of music is the first step to analysis. Depending on the characteristics of the composer and time period different features would be useful. Often features are extracted en masse and then work is done later to determine which features are important or useful in identifying style.

In addition to what kinds of features, there is also the question of the scale at where those features take place. They can be features for a given instrument, the entire piece, or each measure. Also windowing techniques can be used where a “window” is created over a given number of bars or notes, and moves through the whole piece. For each window, a feature is recorded. These can be overlapping windows by creating an “offset” of a number of beats or notes. This produces more data, as instead of one feature for each piece, there is a feature for each window, and there an be tens of windows in each piece.

Common types of features used before in music analysis are: Frequencies or fractions of notes, chords, etc are a common low-level feature. These include the fraction of the score that consisted of dissonant sororities, as well as the fraction of bars that begin with a dissonant sonority. Other features include the type of intervals or consonances present in a piece: perfect consonance, imperfect consonances, and dissonance. In polyphonic pieces, the four types of motion, (parallel, similar, oblique, and contrary) can also be used as features.

Features measuring “stability” are also popular. Stability is computed by dividing the standard deviation of the lengths of the fragment by the mean length of the fragments. It is normalized in this way to be comparable over differing time signatures. These types of features were used by Backer et. al.(Backer & Kranenburg, 2005). They also used overlapping windowing over each entire composition to produce more data. They chose a window of 30 bars to create a high enough number of fragments per piece and a low enough variance of the feature values between fragments. They chose to extract 20 features including features of fractions and measuring stability, and entropy. They computed the entropy of the probability of occurrence of ways of thinking about chords; chords are the same no matter what scale degree they are on, and distinguishing chords differently. Next the entropy was calculated given the probability of each pitch in the score. Entropy was calculated by $-\sum_{i=1}^{N} p_i \log p_i$ where p_i is the probability of occurrence, and N is the total number.Next they the average number of active voices at one time. This represents the voice density of the piece. Then for every interval, the duration of that interval was divided by the total duration of all intervals. Next the total duration of parallel thirds, fourths, and sixths divided by the total duration of all intervals was measured. Finally a measure of suspensions was found.

Brinkman et.al (Brinkman et al., n.d.) used both high level and low level features. The high level features were 9-8 suspensions, oblique motion, contrary motion, similar motion and parallel motion. The low level features were average melodic entropy, normalized pairwise variability index (?), and note-to-note transition probabilities.

Speiser and Gupta (Speiser & Gupta, n.d.) extracted four categories of features, frequencies of individual notes, frequencies of pairwise interval combinations between each of the voices, Markov transition matrices for the rhythms of the pieces, and Markov transition matrices of the pitches in each piece. In total, this lead to a total of 3000 features.

Other features that can be helpful for Renaissance and baroque composers look specifically at differences in counterpoint. Since most composers in that era for the most part followed the rules of counterpoint, these features are created to try to detect specific identifiable uses of counterpoint. These features include counterpoint movement types, dissonance distributions, parallel intervals of each kind, and vertical interval distributions. Mearns used such features (Mearns et al., 2010). They calculated intervals for every subsequent pitch in each part. A weighted count of each interval was stored. They were weighted according to duration to account for the perception of use. The count vector was then normalized to account for the weighted interval content of the score. For movement types, they used ideas based on the rules of first species counterpoint. (Similar, oblique, parallel, and contrary). It does not make musical sense to compare every single adjacent group of notes. Many of the note groups occurring at fractional metrical positions consist of notes held from the previous note group with the addition of a passing or neighbor note in another voice or voices. For this reason, only notes that occur at strong metrical positions, in this case on each beat of the bar, are used for contrapuntal evaluation. This is a relatively simplistic method of choosing structurally important notes. In Bach, for example,the majority of voice progressions take place at a closer level. The counterpoint method is therefore given successive 'on the beat' note groups from which it generates a series of features based on contrapuntal principles applied to voice pairs within note groups. The features include the nature of the approach to a perfect consonance, whether a dissonance is properly prepared (i.e. preceded by a consonant interval containing the same note which then becomes dissonant), whether a dissonance is correctly resolved downwards by step, details of parallel intervals, and the overall distribution of contrapuntal movements (oblique / contrary / similar / parallel / other). The final data set also contains the total vertical interval content as previously described. (reword)

1.6.2 Previous modeling in applications

Most of the previous research has needed to do some kind of feature selection. A lot of features are extracted as a prior i we don't know which features are distinguishing.

A modification of a forward selection (Floating Forward Selection (Pudil, Novovičová, & Kittler, 1994) was used in Backer's (2005) study to extract features in order to identify distinguishing style between Bach, Handel, Telemann, Mozart, and Haydn, and then subsequently classify the authorship of BWV 535. Each composer was compared via creating comparisons of all possible class arrangements, ie (Bach)(Handel), (Bach)(Handel,Teleman), etc. The algorithm extracted features for each class arrangement that distinguished the groups the best. A decision boundary was used for Bach and not Bach, on the features Diss Part, Par thirds, and stab time slice. A k-nearest neighbors classifier was successful in comparing Bach and others as well as each individual composer. Decision trees to interpret the features used in decision making of the different class arrangements. To determine authorship of BWV 535, they trained a quadratic Bayesian classifier to distinguish J.S. Bach, W.F. Bach and J.L Krebs. They again compare every possible class arrangement as potential composers.

PCA was used to analyze the music of Bach, Handel, Telemann, Mozart and Haydn and compare J.S. Bach, W.F. Bach and J.L Krebs in an attempt to classify BWV

534.(Backer & Kranenburg, 2005) Although only two PC's accounted for most of the variance, 5 PCs were used to account for more variance. Binary comparisons were used to compare composers. This resulted in a relatively clear separation between Bach and Josquin. For Josquin and his contemporaries, the PC's do not do as well a job of separation. The results of the principal component analysis run on all the composers, were used to train a classifier on all the composers. First a k-nearest neighbor classifier was used. To account for most of the variance, 27 PCs were used. Next they trained a support vector machine classifier with a radial kernel. Finally they used a decision tree to determine which features were important in discerning the composers.

Speiser and Gupta (Speiser & Gupta, n.d.) scored each feature by the mutual information of each features. They then chose the top 50 features and ran GDA. They then ran PCA to attempt to remove some of the dependencies associated with musical features. They first fit a Naive Bayes for classification, but it had a large training error as the independence assumption does not work well with musical data. Next they used support vector machines with a Gaussian kernel and GCA learning algorithms.

Mearns fit a classifier using a WEKA algorithm, as well as Naive Bayes and a Decision Tree was created that correctly predicted composer 2/3 of the time. (Mearns et al., 2010)

1.7 Fanny and Felix Mendelssohn

Most musical stylometry analysis focuses on music of the Renaissance and Baroque period, as there are more questions of authorship in that period. As the Romantic period is much more modern in comparison, there are many more surviving records of original manuscripts that include the composer.

Felix Mendelssohn, often considered a prodigy akin to Mozart, was a prolific composer. Before he was fourteen years old, he had already written over 100 compositions.

His lesser known sister Fanny Hensel was also a composer of incredible skill. The two were very close, for many years training and studying together. In their early education living in Berlin, Felix and Fanny received the same musical education, first piano lessons by Madam Bigot, a famous pianist esteemed by Haydn and Beethoven. Beginning in 1818, Carl Friedrich Zelter, a somewhat removed student of Bach and the most influential Berlin musician of the time, began to teach them both composition. In addition to music, the children were tutored by some of the finest scholars in Berlin in subjects such as languages, history, and drawing. Goethe himself claimed that Fanny was "as gifted as Felix". (Tillard, 1996)

As Fanny grew up, her father started implying that she should focus her energy on the domestic sphere of her life. While the fact that she never became a world famous composer and performer is often attributed to the gender politics of her time, it is also likely due to her high class. (Reich, 1991) Especially considering the anti-semetic feelings of the time, and since the family had recently converted from Judaism to Christianity, the family did not want any other unusual characteristic such as a professional female composer to set them further apart from "polite" society.

Most of Fanny's available work are Lieder, short pieces of voice accompanied by piano. They were accepted at the time as the more feminine, domestic compositions, acceptable for women to compose. Her brother moved on to more elaborate compositions such as operas and orchestral concertos. Her father pressured Fanny to remain composing Lider. (Todd, 2003)

> "Music will perhaps become his profession, while for you it can and must only be an ornament, never the root of your being and doing. We may therefore pardon him some ambition and desire to be acknowledged in a pursuit which appears very important to him,... while it does you credit that you have always shown yourself good and sensible in these matters; ... Remain true to these sentiments and to this line of conduct; they are feminine, and only what is truly feminine is an ornament to your sex."

Throughout their lives, Felix and Fanny maintained contact through letters until Fanny's death in 1847 and Felix's death shortly thereafter. These letters contain many instances of Felix asking for advice on his compositions (include quote)

Unlike Felix who conducted and performed piano and organ in some of Berlin's most esteemed concert halls, most of Fanny's performances were private, only performed in small circles of her friends and family at intimate parties. Similarly, although she was quite a prolific composer, under recommendation of her father Abraham Mendelssohn, and to a lesser extent Felix, Fanny did not publish her work until later in her life. In 1846 after her fathers death and though her brothers disapproval, she published her first collection of Lieder. Many of Fanny's unpublished notebooks are in private collections and are inaccessible

However, it is widely speculated (known?) that some of Fanny's work was published under her brother's name, Especially three pieces each in his Op 8 and 9 lieder. Famously, when Felix met the Queen of England, she sang Felix's Lied "Italien", and Felix had to admit that in fact, it was his sister that had written it. In a letter to Felix, Fanny admits:

> "I have just recently received a letter from Vienna, which contained basically nothing but the question of whether "On Wings of Song" was by me, and that I should really send a list of things that are running about in the world disguised, it seems that they aren't clever enough themselves to separate the wheat from the chaff." (Mace, 2013)

As she never made such a list, we are left to wonder if there are any other pieces of hers that have been published under her brothers name and reputation.

This project will use Lieder of Fanny and Felix Mendelssohn. Most of the available work by Fanny are Lieder, of which Felix also composed a great deal. We will see if there is a determinable difference in style of these siblings who grew up very close and received mostly the same musical education. We will then look at the (disputed?) Op 8 and 9. Additionally, using Lieder that have been decidedly written by Felix, we will see if any other of his earlier publications could have potentially been written by Fanny.

Chapter 2

About the data and conversion process

2.1 Pieces used

The majority of the pieces used in this paper were Lieder of Felix Mendelssohn and Fanny Hensel. While both composers composed different types of music, the majority of available scores from Fanny Hensel were Lieder. Felix Mendelssohn composed many different styles of music, orchestral, piano, etc. Fanny Hensel in contrast has an available existing corpus of mostly Lieder, although she did compose many works for solo piano and orchestra. Of Felix's music there were a total of 116 pieces: Lieder of Op 8 (12 pieces), op 9 (12 pieces), op 19(6 pieces), op 34 (6 pieces), op 47 (6 pieces), op 57 (6 pieces), op 71 (6 pieces), and 6 pieces of Lied without opus numbers, also 56 Lied without a collection.

Of Fanny's music, a total of 43 pieces were used: 23 lieder were used from her Lieder without Name collection, 10 from her wo kommst du her collection, and 10 from an unnamed collection.

Data from JS Bach was also used. These data were available in Kern Score format. from the Center for Computer Assisted Research in the Humanities. (CCARH). The pieces used were from the Well Tempered Clavier (WTC). These were written as training pieces and each book contains 24 pieces with one in every possible key. These were chosen as the data were more easily accessible (no scanning was required), and they were a similar format as the Mendelssohn songs, written as for solo piano(or harpsichord).

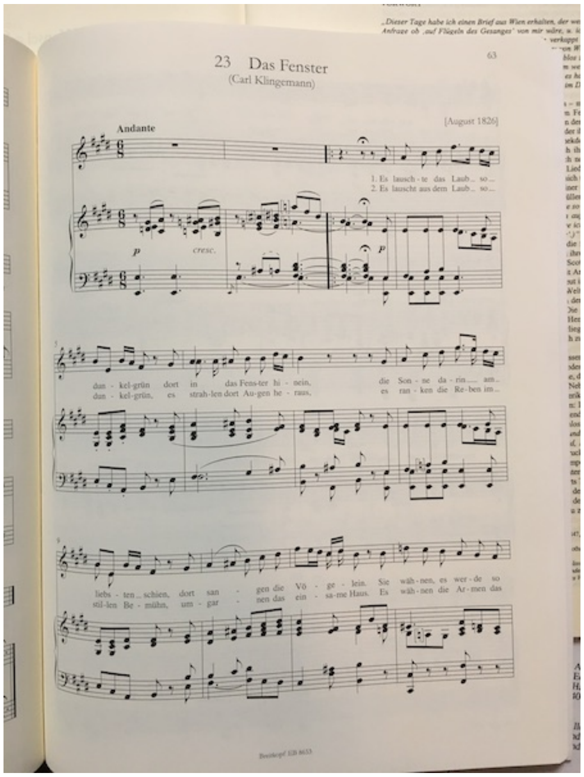

23 Das Fenster
(Carl Klingemann)
63
Andante
[August 1826]
1. Es lausch-te das Laub.. so...
2. Es lauscht aus dem Laub.. so...
p
cresc.
p
dun-kel-grün dort in das Fens-ter hi-nein, die Son-ne da-rin.... am...
dun-kel-grün, es strah-len dort Au-gen he-raus, es ran-ken die Re-ben im...
liebs-ten... schien, dort san-gen die Vö-ge-lein. Sie wäh-nen, es wer-de so
stil-len Be-mühn, um-gar-nen das ein-sa-me Haus. Es wäh-nen die Ar-men das

Figure 2.1: A picture of a score

2.2 Optical music recognition

The vast majority of classical music is found solely in PDF or physical copies. Sheet music as a form of data requires a lengthy process of conversion before being able to be used in any analysis. Simply scanning the scores into, say, a PDF, gives no musical semantics and can only be viewed on screen or printed on paper. Thus, the two main steps in reading in data from sheet music are: using optical music recognition software to transform physical scores into digital formats, and second,to read the digital format in to R where subsequent analysis can be done.

The scores used in this paper were obtained from physical copies available in the Reed music library. These scores were then scanned using software designed for optical music recognition (OMR). Optical music recognition requires learning from graphical and textual information. The main things the software must pick up are the locations of bar lines, notes, rests, slurs, dynamic markings, tempo markings, lyrics etc. Basic optical music recognition has been around since 1966.

Most commonly, the first step in optical music recognition is to remove the staff lines. The staff lines are critical, as they define the basis for the vertical definition distance of pitch, and the horizontal distance definition of rhythm. The staff gives a normalization that is helpful, essentially defining the size of what notes and rhythm look like. Staff removal methods include projections, histograms, run lengths, candidates assemblage, contour tracking, and graph path search. (Doermann, Tombre, & others, 2014)

The next step is music symbol extraction and classification. These methods include template matching, where the object in question is compared to existing known musical symbols, simple operators, such as analysis of bounding boxes and projections, joining graphical primitives, such as combining extracted objects such as notes, note heads, and note beams to connect them in a musically correct way to form chords etc. Other methods use statistical models for analyzing musical primitives (the objects its trying to classify) such as Neural Networks, Support Vector Machines, k-Nearest Neighbor, and Hidden Markov Models.

The next step OMR preforms is syntactical analysis and validation. This step essentially uses defined grammars describing the organization of music notation in terms of music symbols. This makes the classification problem simpler, as there are existing rules and relationships between musical symbols.

```
knitr::include_graphics("images/museScore.png")
```

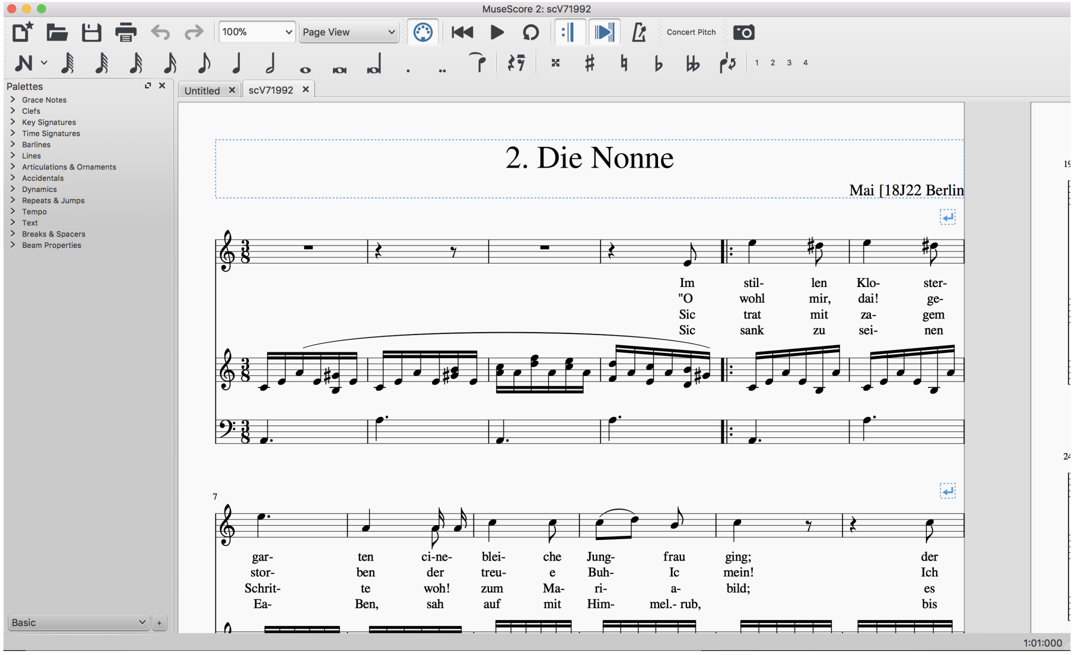


Figure 2.2: A score in MuseScore format after being read by PhotoScore

The OMR used in this paper was Photo Score. Photo Score works by scanning the score on a flatbed scanner at a high resolution. It then uses OMR techniques to output a musicXML file that can be read in by most music composing software, such as Sibelius, Finale or Muse Score. MusicXML is commonly used as a format for digital music, as it is conducive to representing sheet music and music notation, and it can be transferable to many different music software. Muse Score was chosen to be the music software for viewing digital scores, as it is a free software that can read MusicXML.

After being scanned by PhotoScore and read into Muse Score, each piece was proof-read and corrected. This involves looking through every piece line by line for each bar to “spell check” the digital version. PhotoScore did a very impressive job recognizing notes, but often had issues recognizing rhythms. There were on average approximately 6 issues per piece. Unfortunately, the scanning process is very lengthy and time consuming, as the scanning often gives a large number of mistakes. Often the score must be then scanned again. In addition the proof-reading process is lengthy. We must check each note and theme for errors against the original score, and change the afflicted notes using Muse Score. The corrected score must then be output as a musicXML file. In addition, there were some pieces that PhotoScore had a hard time reading. These pieces were then entered into MuseScore by hand and then proofread.

MusicXML on its own is not conducive to converting into a data frame as representing the single half note middle C looks like this:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC
    "-//Recordare//DTD MusicXML 0.5 Partwise//EN"
    "http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise>
  <part-list>
    <score-part id="P1">
      <part-name>Music</part-name>
    </score-part>
  </part-list>
  <part id="P1">
     <measure number="1">
      <attributes>
        <divisions>1</divisions>
        <key>
          <fifths>0</fifths>
        </key>
        <time>
          <beats>4</beats>
          <beat-type>4</beat-type>
        </time>
        <clef>
          <sign>G</sign>
          <line>2</line>
        </clef>
      </attributes>
      <note>
        <pitch>
          <step>C</step>
          <octave>4</octave>
        </pitch>
        <duration>4</duration>
        <type>whole</type>
      </note>
    </measure>
  </part>
</score-partwise>
```

We then need to convert into a format more easily readable into R. The Kern Score music format is much more easily readable. It has clearly expressed time signature, bar, beat and musical voicing information (Mearns et al., 2010)(mearns). Because of this, it is also more conducive to being read into an R data frame. The below picture shows how a basic piece of music corresponds to a .krn file.

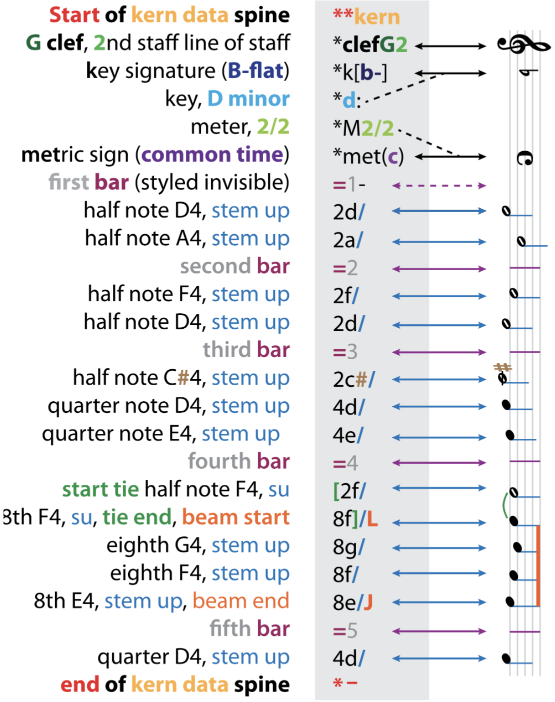


Figure 2.3: How sheet music corresponds to kern files

Kern files are organized with columns each representing one line of music. Each line of a Kern file represents one note of one value of a time base. The time base for a Kern file is based on the smallest (shortest) rhythm value of a note found in a piece. For example, if a piece was in 4/4 and there were sixteenth notes present there would be 16*4 rows for each measure. The "attack" of each note is the only note printed, the following time while the note is held is represented with dots in the remaining rows until a new note is sounded for that staff. The pitch of each note is represented by the letters a through g. The case (lower or upper) as well as the repetition (c or ccc) represents which octave the pitch occurs. Any accidental is represented with a #, -, or n symbol. Each instrument/staff in a piece is represented using one (or more) columns called splines. For example, in most Lied, there is voice and piano. There are thus three splines, one for voice, one for the treble clef staff of the piano, and one for the bass clef of the piano. In addition, there are splines that contain the text for the voice for the corresponding notes. This was not of interest to the musical classification problem, so these splines were removed. Chords are represented by multiple notes on the same line. For example, if there was a half note C major triad followed by a quarter note D flat, it would be represented as

```
2c 2e 2g
4d- . .
```

In addition, there is a lot of information about the appearance of the piece, stem direction etc for notes, but this was decided as not important for determining style, so it was removed.

We convert to .krn format by using Humdrum's function xml2hum that converts a musicXML file into a .krn file. Humdrum is a computational music software used to analyze music. It is a command line tool that has many functions for music analysis. The Kern file type can be read much more easily into R. Compared to above, the code for a single middle c whole note would be :

```
**kern
*clefG2
*k[c]
*M4/4
=1-
1c/
```

The import_xml_files.sh file goes through the process of converting scores from musicXML to .krn. Each spline needs to to be individually converted using xml2hum. The individual splines are then converted into the same time base (there are issues when the bass line only has half notes and the soprano line has a lot of 32nds). This conversion essentially adds dots as placements so that the splines can line up correctly by measure and beat.

The CCARH has a large data base mainly baroque and Renaissance composers already in the Kern format, which is where the Bach data came from. The files that were scanned need to be separated into having a separate file for each staff, which would mean a seperate file for each instrument. In addition, since we are focused on musical style, the text of the pieces are removed in this stage. In our case for each piece there are always two or three files for each piece, which are voice, piano right hand, and piano left hand. This is necessary to avoid the bugs in xml2hum that have issues when staffs don't necessarily match up as a result of the conversion process, most often by rhythm.

2.3 .krn to R

Once we have .krn files to represent each piece we use regular expressions to extract key information. For scanned music (Felix and Fanny music), there are as many files are there are staffs, usually three. MuseR's `krn2df()` and `piece_df()` functions read in .krn files and output a data frame in R for each piece. First the data in .krn format are read in line by line using R's `readLines()` function. This takes every line of the .krn file and converts it into a vector. Each entry contains the rythem value and note value for all notes in that line. If there are multiple notes played at the same time, they are all in one line. The notes are separated by splitting up the string by spaces. This converts a single string representing one Kern line to multiple strings each representing one note (or dot placeholder) for one Kern line. Then each entry is separated out into the theme and note value for each note. Each line contains the following columns: the measure the note occurs in, the rhythm value for the (for example 4), note name, octave inclusive (for example cc), note name (octave exclusive)[C sharp]. In addition, for the whole piece the key signature and meter are recorded as columns. If there are 3 splines and each spline has at most one note at a time, there would thus be $3 + 3 * 3 = 12$. If there are 3 splines and one of the splines has at most 3 values, that is equivalent to having 5 total splines there are then $ 3 + 3*5 = 18$ columns.

A lot of data included in the .krn files are not necessary. For example, we assume that whether or not a note has a step up or stem down offers no help in classifying composer style, so this information is removed when converting to an R data frame.

Inspired by the .krn file type, each row of the R data frame contains one time base value. For a given piece, the time base represents the shortest note duration value. For example, if the shortest note a piece contained was a sixteenth note, the time base would be 16. Each measure then would contain 16 rows. This results in many rows of NA for certain instruments, when a note is still being voiced, but it is not the instance of the note being attacked.

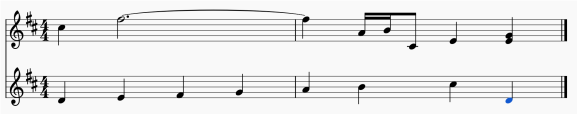
Chapter 3

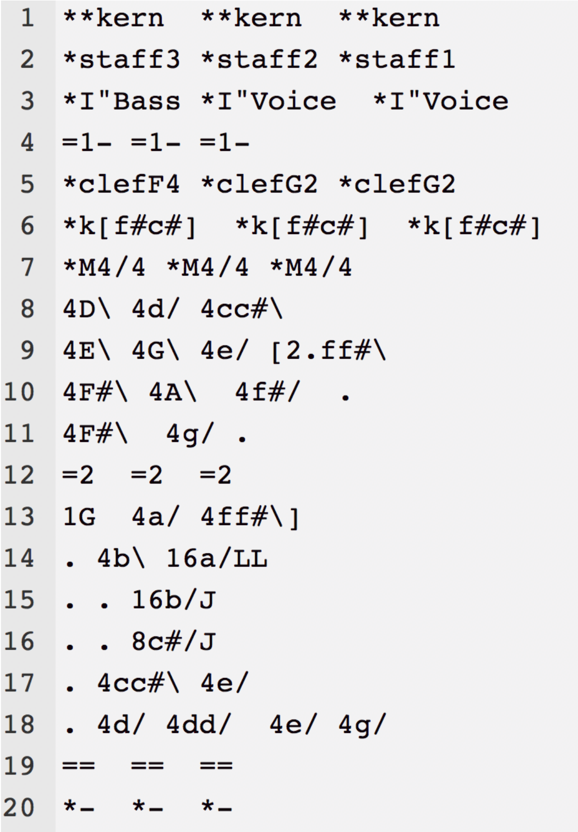
MuseR and features

To the best of my knowledge, there is currently no package of R that has been built to analyze sheet music. There are existing packages (such as tuneR) that examine audio formats of music. The intention of this thesis was to create a package, museR, that takes sheet music in the proper form (musicXML or .krn) and does all of the analysis using R.

3.1 Reading in data

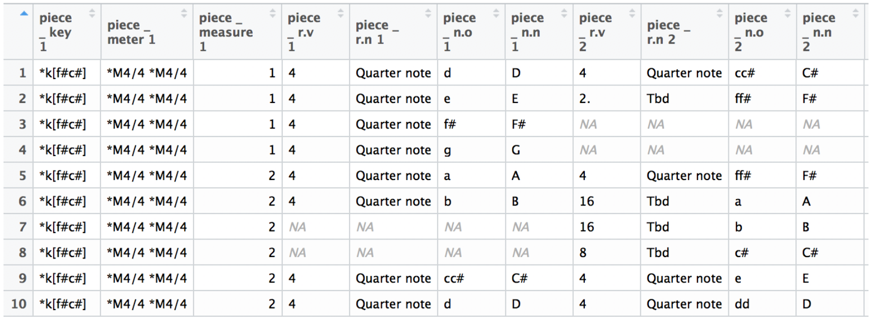
MuseR is equipped to take in data in the .krn format. The functions for this are `kern2df()` and `piece_df()` These are most usefully in the form of individual splines. This allows for naming the columns according to instrument. If the following piece is represented in .krn format by





```
**kern   **kern   **kern
*staff3 *staff2 *staff1
*I"Bass *I"Voice   *I"Voice
=1- =1- =1-
*clefF4 *clefG2 *clefG2
*k[f#c#]   *k[f#c#]   *k[f#c#]
*M4/4 *M4/4 *M4/4
4D\ 4d/ 4cc#\
4E\ 4G\ 4e/ [2.ff#\
4F#\ 4A\   4f#/   .
4F#\   4g/ .
=2   =2   =2
1G   4a/ 4ff#\]
. 4b\ 16a/LL
. . 16b/J
. . 8c#/J
. 4cc#\ 4e/
. 4d/ 4dd/   4e/ 4g/
==   ==   ==
*-   *-   *-
```

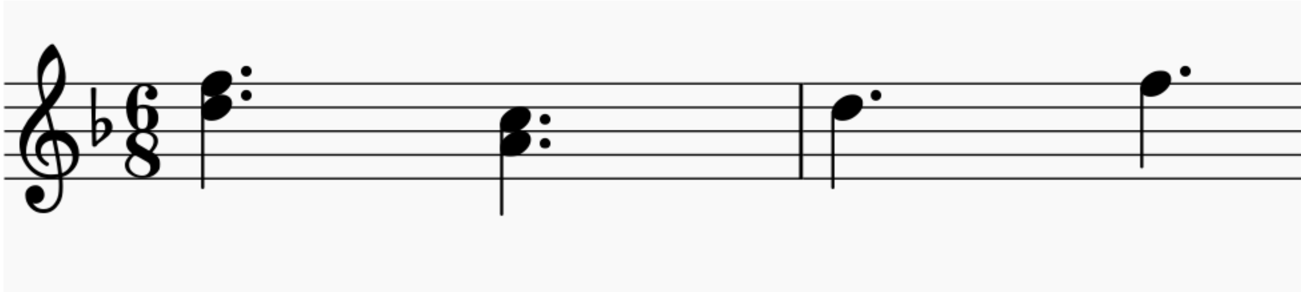
`piece_df()` outputs the following data frame



	piece_key_1	piece_meter_1	piece_measure_1	piece_r.v_1	piece_r.n_1	piece_n.o_1	piece_n.n_1	piece_r.v_2	piece_r.n_2	piece_n.o_2	piece_n.n_2
1	*k[f#c#]	*M4/4 *M4/4	1	4	Quarter note	d	D	4	Quarter note	cc#	C#
2	*k[f#c#]	*M4/4 *M4/4	1	4	Quarter note	e	E	2.	Tbd	ff#	F#
3	*k[f#c#]	*M4/4 *M4/4	1	4	Quarter note	f#	F#	NA	NA	NA	NA
4	*k[f#c#]	*M4/4 *M4/4	1	4	Quarter note	g	G	NA	NA	NA	NA
5	*k[f#c#]	*M4/4 *M4/4	2	4	Quarter note	a	A	4	Quarter note	ff#	F#
6	*k[f#c#]	*M4/4 *M4/4	2	4	Quarter note	b	B	16	Tbd	a	A
7	*k[f#c#]	*M4/4 *M4/4	2	NA	NA	NA	NA	16	Tbd	b	B
8	*k[f#c#]	*M4/4 *M4/4	2	NA	NA	NA	NA	8	Tbd	c#	C#
9	*k[f#c#]	*M4/4 *M4/4	2	4	Quarter note	cc#	C#	4	Quarter note	e	E
10	*k[f#c#]	*M4/4 *M4/4	2	4	Quarter note	d	D	4	Quarter note	dd	D

Melodic intervals

Melodic intervals, or the interval between two successive notes, are found using the `mel_ints()` function. This first calculates the top line of any instrument. This then outputs the proportion of each melodic interval happening over the whole piece. There are 12 possible intervals that are counted (ignoring augmented and diminished): unison,m2,M2,m3,M3,p4,tt,p5,m6,M6,m7,M7. `mel_ints()` outputs a vector of the proportion of each of the intervals.



For example if this function was run on the above piece, the melodic top line intervals would be: $\{(f,c),(c,d),(d,f)\} = \{p4, M2, m3\}$, which would output a proportion of: $(0, 1/3, 1/3, 0, 0, 1/3, 0, 0, 0, 0, 0, 0)$

Similarly the `connsonance()` function outputs the proportion of consonant (perfect, imperfect, dissonant) intervals over the whole piece by calling `mel_ints()`.

Density

Major_minor

For most musical analysis, the key of the piece is important in determining chords, etc. The key is based on the key signature, which is always given in a .krn file. Kern files from CCARH have the key of the piece given, but scanned files do not. For kern files from CCARH, `Major_minor()` extracts the key given by the .krn file. for scanned files, `Major_minor()` identifies the two options for key given the key signature. For example, if there was a key signature with one sharp, the options would be G Major or E minor. The tonic for each option is identified, and then the count of instances of both choices for tonic is made. The key is determined by which of the options for tonic has the higher count. When this was tested on the WTC pieces, it correctly classified x/48 of them. (I think I should add some weighting for the first and last note in the future..)

Prop scale degree

Chords

Suspended chords are currently not supported by MuseR. Chords that begin, or are “attacked” at the same time count.

First, the key of the piece is found, as different chords depend on the key.

Next, the times notes are played at the same time are extracted into a list. Then the number of unique notes played at once is found. If there are two notes played at once, `harm_int()` calculates the harmonic interval. This is done by calculating

$$note_1 - note_2 \mod (12)$$

This gives the number of half steps between each note. That number is then matched with the index of the interval. Work is being done to have this include augmented and diminished interval, but unfortunately that has not been completed at this time.

The possible triad chords are all defined by the intervals between each note. For example, a Major triad is given by the base note, a major third above the base, and a perfect fifth above the base. This corresponds to 4 half steps then 4 half steps. Alternatively a minor triad is given by the base, a minor third above, and a perfect fifth above the base, which is 3 half steps, then 5 half steps.

A similar process is done for seventh chords.

Resolutions

Par thirds, par fourths, par sixths

Chapter 4

EDA and Feature Selection

Chapter 5

Models

For classification, we are interested in using a set of features to predict the response, or composer. We denote the features X, where X_i is a certain feature. Each song has a composer or response, known or unknown, denoted by Y. The i^{th} song has features x_i and composer Y_i, where x_i is a vector of p features. The composer takes values in a discrete set, which is the possible composers. Thus we can always divide the input space into a collection of regions labeled according to the classificatoin

5.1 Linear Methods for Classification

5.1.1 Linear Regression:

A naive model for classification is linear regression. If our predictor space G has K classes, we code the response K different indicator responses Y_k where $Y_k = 1$ if $G = k$ and 0 otherwise. We can use the resulting k hyperplanes as a decision boundary. We find the coefficients for the line by finding coefficients β to minimize the residual sum of squares.

$$RSS(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$$

where $\mathbf{X}$ is an $N \times p$ matrix with each row an input vector, and $\mathbf{y}$ is an N-vector of the outputs of the training set. This gives the unique solution:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

This gives us;

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

If there are K classes, and we have that the fitted linear model for the kth class is $\hat{f}_k(x) = \hat{\beta}_{k0} + \hat{\beta}_k^T x$, the decision boundary between class k and class l is the set of points for which $\hat{f}_k(x) = \hat{f}_l(x)$ which is equivalent to the set $\{x : (\hat{\beta}_{k0} - \hat{l}0) + (\hat{\beta}_k - \hat{\beta}_l)^T x = 0\}$ which is the hyperplane.

Linear Discriminant Analysis

Linear regression on a categorical variable that has multiple variables has issues when there isn't a natural ordering with the categories. For large K and small p, groups can be masked. When there is a binary response, we can calculate $P(Y|X)$, but linear regression can give predictions that arent valid probabilities, namely negative probabilities or probabilities greater than 1.

Knowing the class posteriors $P(Y = k|X)$ gives us an optimal classification. If we assume $f_k(x)$ is the class-conditional density of X in class $G = k$ and that π_k is the prior probability of class k with $\sum_{k=1}^K \pi_k = 1$. We can then model $P(Y = k|X)$ by modeling the distribution of the features X seperately in each response class, and then use Bayes' theorem to calculate $P(Y = k|X)$ which gives us the following:

$$P(Y = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^K f_l(x)\pi_l}$$

We thus must have a model to find $f_k(x)$. Linear and quadratic discriminant analysis assume a multivariate Gaussian density, given by:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\mathbf{\Sigma}_k^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T\mathbf{\Sigma}_k^{-1}(x-\mu_k)}$$

Linear discriminant analysis (LDA) assumes that the covariance matrix is equal for every k: $\mathbf{\Sigma}_k = \mathbf{\Sigma}\forall k$. Quadratic discriminant analysis does not have this assumption. In addition we assume $\hat{\pi}_k = N_k/N$ where N_k is the number of class - k observations, $\hat{\mu}_k = \sum_{g_i=k} x_i/N_k$, and $\hat{\mathbf{\Sigma}} = \sum_{k=1}^K \sum_{g_i=k}(x_i - \hat{\mu}_k)(x_i - \hat{\mu}_k)^T/(N - K)$

For LDA we can lok at the log ratio comparing two classes k and l and can show:

$$\log\frac{P(Y = k|x = x)}{P(Y = l|X = x)} = \log\frac{f_k(x)}{f_l(x)} + \log\frac{\pi_k}{\pi_l} = \log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T\mathbf{\Sigma}^{-1}(\mu_k - \mu_l) + x^T\mathbf{\Sigma}^{-1}(\mu_k - \mu_l)$$

This is a linear equation, so the classes will be seperated by hyperplanes. From the above, we can find that the predicted class for any x is :

$$\delta_k(x) = x^T\mathbf{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\mathbf{\Sigma}^{-1}\mu_k + \log\pi_k$$

These functions are known as *linear discriminant functions* We predict the class by finding the maximum value of the discriminant funcitions of all k.

For QDA we get the following discriminant functions:

$$\delta_k(x) = -\frac{1}{2}\log|\mathbf{\Sigma}_k| - \frac{1}{2}(x - \mu_k)^T\mathbf{\Sigma}_k^{-1}(x - \mu_k) + \log\pi_k$$

Linear discriminant analysis is helpful when the classes are well seperated, when n is small and the distribution of the predictors X is approximately normal in each of the classes, and when there are more than two response classes.

Logistic Regression

Logistic regression differes from linear discriminant analysis by directly modeling $P(Y = k|X)$ by using the logistic function. The idea is to model the posterior probabilities of each of the K classes as linear functions in x and requiring that the probabilities sum to 1. The model has the form:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

which can be written as:

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

We estimate the regression coefficients by using maximum likelyhood. The log-likelihood for N observations is:

$$\ell(\theta) = \sum_{i=1}^{N} \log p_{g_i}(x_i;\theta)$$

where $p_k(x_i;\theta) = P(Y = k|X = x_i;\theta)$ We then choose θ to maximize this function.

5.2 K- nearest - neighbor

Another method for classification is k-nearest neighbor methods. It uses observations in the training set closest to x to form $\hat{Y}$, the outputs. We often use Euclidean distance as a metric for closness, although other methods exist. It is defined as

$$\hat{Y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

where $N_k(x)$ is the k closest points, or neighbors, x_i in the training set. This is equivalent to taking the average of k observations with x_i closest to x. This gives a predicted class of taking the mode of the k nearest neighbors.

Initial approaches often use linear methods for classification. If we are trying to predict the author, $y \in \{\text{Fanny}, \text{Felix}\}$, or more generally, $y \in \{\text{list of composers}\}$, given features or predictors $\mathbf{X} = \{X_{i \cdots X_p}$, we can divide the input space* into a collection of regions labeled according to the classification.

We can create linear decision boundaries for K classes where the fitted linear model for the kth indicator response is $\hat{f}_k(x) = \hat{\beta}_{k0} + \hat{\beta}_k^T x$. The decision boundary between classes i and j is the set of points for which $\hat{f}_i(x) = \hat{f}_j(x)$, or in other words, the set $\{x : (\hat{\beta}_{i0} - \hat{\beta}_{j0}) + (\hat{\beta}_i - \hat{\beta}_j)^T x = 0\}$ which defines a hyperplane.

Similarly quadratic decision boundaries can be used when we increase our predictor space to include squares and higher polynomials of X. We then fit linear decision boundaries, which then map down to quadratic functions in the original space.*.

Logistic regression is often used when the response is binary. It models the probability that Y belongs to either category. We use the logistic function $p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$ To calculate estimates of $\hat{\beta}_0, \hat{\beta}_1$, we use maximum likelihood. To do this, we choose $\hat{\beta}_0, \hat{\beta}_1$ to maximize the function ...

The Naive Bayes classifier is often used for musical classification as it is good when the dimension p of the features space is large, makeing density estimation unattractive*. It makes the (naive) assumption that all the features are independent for a given class i.

$$f_i(X) = \prod_{k=1}^{p} f_{ik}(X_k)$$

The idea of separating Hyperplanes is essential to Support Vector Machines (SVM)

Chapter 6

Model Fit

6.1 Model Fit

Chapter 7

Discussion

7.1 Discussion

Conclusion

If we don't want Conclusion to have a chapter number next to it, we can add the `{-}` attribute.

More info

And here's some other random info: the first paragraph after a chapter title or section head *shouldn't be* indented, because indents are to tell the reader that you're starting a new paragraph. Since that's obvious after a chapter or section title, proper typesetting doesn't add an indent there.

Appendix A

The First Appendix

This first appendix includes all of the R chunks of code that were hidden throughout the document (using the `include = FALSE` chunk tag) to help with readibility and/or setup.

In the main Rmd file

```r
# This chunk ensures that the thesisdown package is
# installed and loaded. This thesisdown package includes
# the template files for the thesis.
if(!require(devtools))
  install.packages("devtools", repos = "http://cran.rstudio.com")
if(!require(thesisdown))
  devtools::install_github("ismayc/thesisdown")
library(thesisdown)
```

In Chapter ??:

References

Adair, D. (1944). The authorship of the disputed federalist papers. *The William and Mary Quarterly: A Magazine of Early American History*, 98–122.

Backer, E., & Kranenburg, P. van. (2005). On musical stylometry—a pattern recognition approach. *Pattern Recognition Letters*, *26*(3), 299–309.

Brinkman, A., Shanahan, D., & Sapp, C. (n.d.). Musical stylometry, machine learning, and attribution studies: A semi-supervised approach to the works of josquin.

Doermann, D., Tombre, K., & others. (2014). *Handbook of document image processing and recognition*. Springer.

Ford, P. L., & Bourne, E. G. (1897). The authorship of the federalist. *The American Historical Review*, *2*(4), 675–687. Retrieved from `http://www.jstor.org/stable/1833983`

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, *3*(Mar), 1157–1182.

Mace, A. R. (2013). *Fanny hensel, felix mendelssohn bartholdy, and the formation of the" mendelssohnian" style* (PhD thesis).

Mearns, L., Tidhar, D., & Dixon, S. (2010). Characterisation of composer style using high-level musical features. In *Proceedings of 3rd international workshop on machine learning and music* (pp. 37–40). ACM.

Mosteller, F., & Wallace, D. (1964). *Inference and disputed authorship: The federalist*. Addison-Wesley.

Papadopoulos, G., & Wiggins, G. (n.d.). AI methods for algorithmic composition: A survey, a critical view and future prospects.

Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, *15*(11), 1119–1125.

Reich, N. B. (1991). The power of class: Fanny hensel. *Mendelssohn and His World*, 86–99.

Speiser, J., & Gupta, V. (n.d.). Composer style attribution. *Project Report for CS*, *229*.

The josquin research project. (n.d.). *The Josquin Research Project*. Retrieved from `http://josquin.stanford.edu/`

Tillard, F. (1996). *Fanny mendelssohn*. Hal Leonard Corporation.

Todd, R. L. (2003). *Mendelssohn: A life in music*. Oxford University Press.