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

Classification for authorship of written classical music is a relatively infrequently explored field. In theory composers leave behind unconsious (or perhaps consious) signals that indicate a piece of music as thier own. Ideally these signals, or features can be extracted and used to build a model to predict the composer of a piece of music. For instance the siblings Fanny Hensel and Felix Mendelssohn, although very similar in compositional style, likely have featuresthat distingush their music. It is speculated that there are at least ... pieces that Fanny wrote that were published by Felix. Low level features, including frequencies of chords and scale degree usage were extracted from ... pieces known to be written by Felix and ... pieces known to be written by Fanny. These features were extracted using museR, an R package written by me*. Additionally, to check the validity of the model, pieces from J.S. Bach's Well Tempered Clavier were used to compare against Felix and Fanny. A lasso logistic model was then trained on Bach and Mendelsssohn data using 5-fold cross validation with a missclassification rate of ... . A lasso logistic model was then trained in the same way on Felix and Fanny data with a missclassification rate of ... Finally the 12 disputed pieces were checked and ...

Chapter 1

1.1 Introduction:

The purpose of this thesis is to use musical stylometry to quantify the difference between the compositions of the siblings Felix and Fanny Mendelssohn. (I hate starting things help. . .)

Treating text or music as data has challenges. Neither contains anything immediately analyzable. With the increasing availability of the internet and social media data, text classification has become an promising form of data analysis. While text classification problems are very frequent, similar methods that classify music have not been as commonly explored. To exploret these problems, first analyzable features must identified that can be extracted for every text. These features can include word frequencies, length etc. Analysis of text using machine learning can find patterns, authors, or categories of text. Similar methods can also be used to classify music.

Feature extraction is an essential part in text or music classification. 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 extracted features are used to build a model that can correctly classify a likely composer for that piece of music. Studies that examine the style of a type of music or composer are known as musical stylometry studies. For musically trained humans, this might be an easy task. Some are able to automatically distinguish a piece composed by Bach or Mozart either by listening to a recording, or looking at sheet music. This becomes more difficult when composers are contemporaries. Mozart and Salieri might be distinguishable to a more discerning listener or scholar, but it might be harder. Harder still is when the identity of the composer piece of music is disputed. Examples of such disputed authorship exist throughout music history, most notably in the Renaissance.

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

1.2 Brief history of text classification

The Federalist Papers were one of the earliest instances of text classification for determining authorship.(Mosteller & Wallace, 1964). The famous Federalist Papers were written under the pen name 'Publius'. There are several of these 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 positions (Adair, 1944). Prior to quantitative analysis, historians had often examined the papers using styles of previously known writings of Madison and Hamilton. Their analysis was 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, Mosteller and Wallace (1964) used the frequency of words such as 'and', 'by', 'from', and 'upon', to train the writings on a set of known writings by each author. These unconscious indicators were able to differentiate between the two writers, and 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 accomplished by reading one word after another. Information in a 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 cadential 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, one can end up with many features to describe one piece. It is possible that the number of predictors (p) can be greater than the sample size (n), or that many features encode essentially the same thing. 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 classical musical stylometry papers have focused on composers in the Renaissance(1400-1600), Baroque(1600-1750), and Classical(1730-1820) eras. The Mendelssohns were composing in the Romantic era(1780-1910). The focus of earlier papers may have been because composers in earlier eras followed rules of counterpoint more exactly, or perhaps had less "expressive" allowances for their composing, thus making features easier. 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, the initial step of coding the music is the most time consuming, having music in a format immediately conducive to analysis is very helpful.

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. Chords occur when there are more than two notes being played simultaneously. The type of chord is also determined by the harmonic intervals it contains. 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. The key signature of the piece indicates how many sharps or flats are normal for the piece. A piece's key can be major or minor, although the piece can internally modulate between the two. The scale degree of a note depends on its position of the key that it is in. The time signature of a piece determines the number of beats that are in each measure. 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". These are also known as sonorities, and divided into three types. Imperfect sonorities are... perfect sonorities are ... and and dissonant sonorities are..

For classical music there are some "rules" that most composers follow to some extent. These were followed more strictly in the Baroque era than the Romantic era. These rules are known as counterpoint. First species counterpoint is a type of counterpoint. One rule is that there are no parallel perfect consonances. Perfect consonances are octaves, fourths and fifths. Parallel indicates that they occur one after another. Another rule is that from one perfect consonance to another, the notes must proceed in contrary or oblique motion.

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. Windowing 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. High level features often depend on the context of where the features was extracted from. For example, cadences and modulations (brief changes in key) throughout the piece are easily found when a human analyzes a piece of music, but are very hard to generalize as a feature. If one wants to use high-level features, they would first have to train the feature to see if it was performing accurately. For this reason, this paper focuses only on low level pieces, or pieces that are context independent.

Especially in research regarding 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 one included every variable that was extracted from a piece, the model would very likely be overfit. Thus some feature selection is important before fitting a model.

1.6 Background on Variable and Feature Selection

Before analysis is done, numerous features are often extracted from the music without knowing a priori 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 and existing variable selection algorithms can help with this process.

Variable ranking uses a score function to assign a score to each possible variable. It is a computationally efficient method and is robust against overfitting 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.

Single variable classifiers rank the variable according to its individual predictive power. The predictive power can be measured in terms of error rate, or using the false positive or false negative rate. This classifier cannot distinguish variables that perfectly separate 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 filters, wrappers, 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 resulting performance from training a given features subset. Embedded methods perform variable selection in the process of training a model 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 successful in making our predictors. The first is an unsupervised problem. The second is supervised.

1.7 Previous research.

There have been numerous other applications of machine learning to music. For example, a previous study 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.)

Research on musical stylometry has focused on two types of data: audio and sheet music. This 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 a model can be fit to a testing data set to predict composer. If the fitted model shows good predictions, that model can be used on the pieces of unknown authorship. Musical stylometry can be used to resolve disputed authorship, but also to detect distinguishing musical styles of composer, even if there are no disputed pieces. This same process can also be used to look at the development of a composers musical style over time.

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.) uses machine learning to evaluate attribution of compositions by Josquin des Prez. They first compared music of Josquin and JS Bach's four part chorales. 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 to find differences in style.

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 b Ockeghem, 183 by 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, 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)

Mearns et. al. (Mearns, Tidhar, & Dixon, 2010) extrected 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 compositional type.

1.8 Previous choices of features in applications

Deciding on and extracting features of music is the first step of 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 of those features. 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 can 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 sonorities, 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 chords. They measured this two different ways, defining chords to be the same no matter what scale degree they are on, and distinguishing chords differently depending on scale degree. 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 calculated average number of active voices at one time which 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 distinguishing 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 features for counterpoint method 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). A feature for total vertical interval count is also used.

1.9 Previous modeling in applications

Most of the previous research has needed to do some kind of feature selection. Usually features are extracted before any analysis is done, and we dont know before hand 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)

1.10 Fanny and Felix Mendelssohn

Most musical stylometry analysis has focused 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 née Mendelssohn 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-Semitic 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, orchestral concertos and symphonies. Her father pressured Fanny to remain composing Lieder. (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.

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 father's death and though her brother disapproved, she published her first collection of Lieder. Many of Fanny's unpublished notebooks are in private collections and are inaccessible for study.

However, it is widely speculated 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. It is suspected that most of Fanny's friends and family would have known at the time that she had composed the pieces published by felix. An article written in the Harmonicon in 1830, an influential music journal in London, mentioned that all the lied in Felix's collection were not written about him:

> " I possess twelve published songs under Mr. Mendelssohn's name, which he wrote when a boy of fifteen. One of these appears in the musical annual, "Apollo's Gift," and deserves all the praise you have in your review bestowed on it. But the whole of the twelve are not by him: three of the best are by his sister, a young lady of great talents and accomplishments. I cannot refrain from mentioning Miss Mendelsshon's name in connexion with these songs, more particularly when I see so many ladies without one atom of genius, coming forward to the public with their musical crudities, and, because these are printed, holding up their heads as if they were finished musicians. Miss Mendelssohn is a first-rate piano-forte player, of which you may form some idea when I mention that she can express the varied beauties of Beethoven's extraordinary trio in Bb, beginning She has not the wild energy of her brother; but possesses sufficient power and nerve for the accurate performance of Beethoven's music. She is no superficial musician; she has studied the science deeply, and writes with the freedom of a master. Her songs are distinguished by tenderness, warmth, and originality: some which I heard were exquisite."(Ayrton, 1830)

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. All other previous composership studies cannot be adjusted for training and upbringing, so it might be challenging to detect a difference.

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. 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 were 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 collection contains 24 pieces with one in every possible key. Pieces from the Well Tempered Clavier 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).

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: first 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

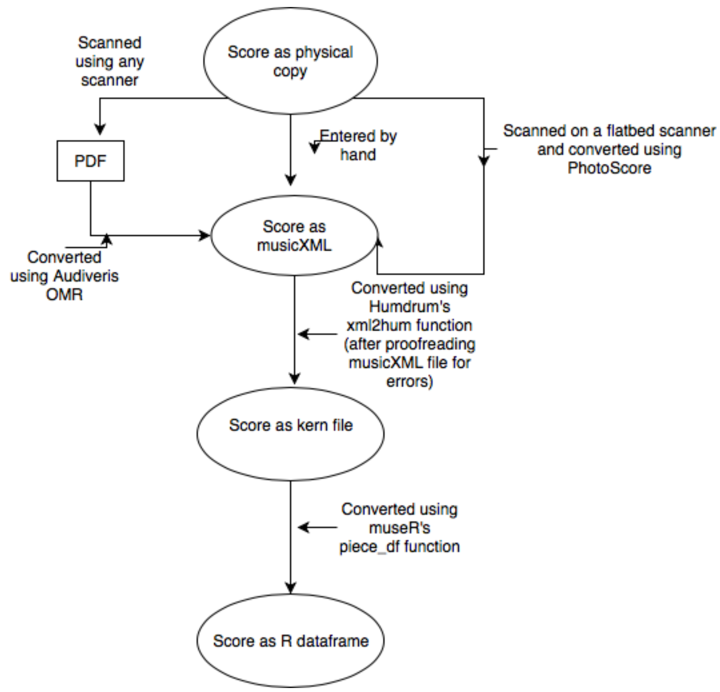

Figure 2.1: Flowchart of the conversion process from physical copy to dataframe

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 the OMR is trying to classify) such as Neural Networks, Support Vector Machines, k-Nearest Neighbor, and Hidden Markov Models.

The next step OMR performs is syntactical analysis and validation. This step

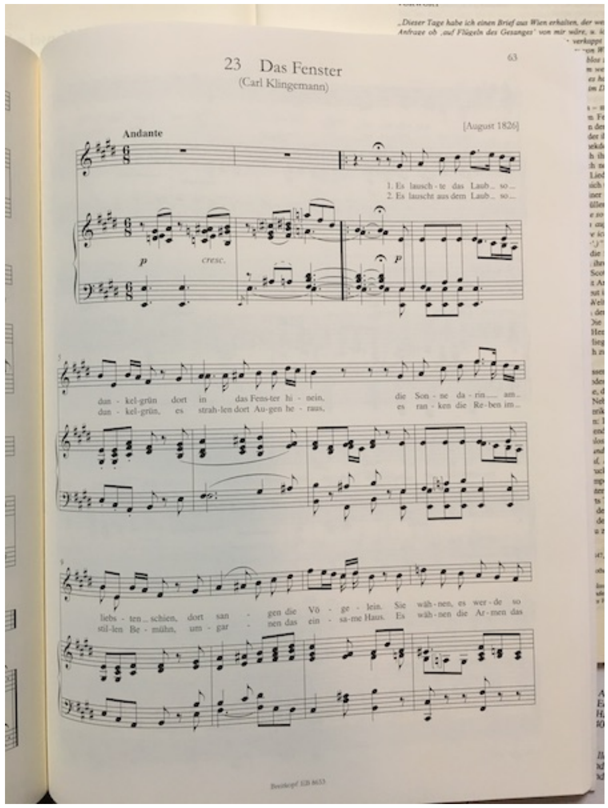

Figure 2.2: Score in physical form before conversion process has started

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.

The two OMR softwares used in this paper were PhotoScore and Audiveris. Each has its own benefits and issues. Photo Score works by scanning the physical 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.

Audiveris in contrast works by inputting a high resolution PDF and then uses OMR techniques to output a music XML file. Often, high enough resolution PDFs do not exist, so the physical scores must also be scanned by any garden variety scanner. 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 proofread and corrected. This involves looking through every piece line by line for each bar to "spell check" the digital version. PhotoScore did a good job recognizing notes, but often had issues recognizing rhythms, and had issues keeping the structure of the piece. Often in the scanning process clefs or bar lines were not found, causing PhotoScore to output every staff on one line. Using PhotoScore, there were on average approximately

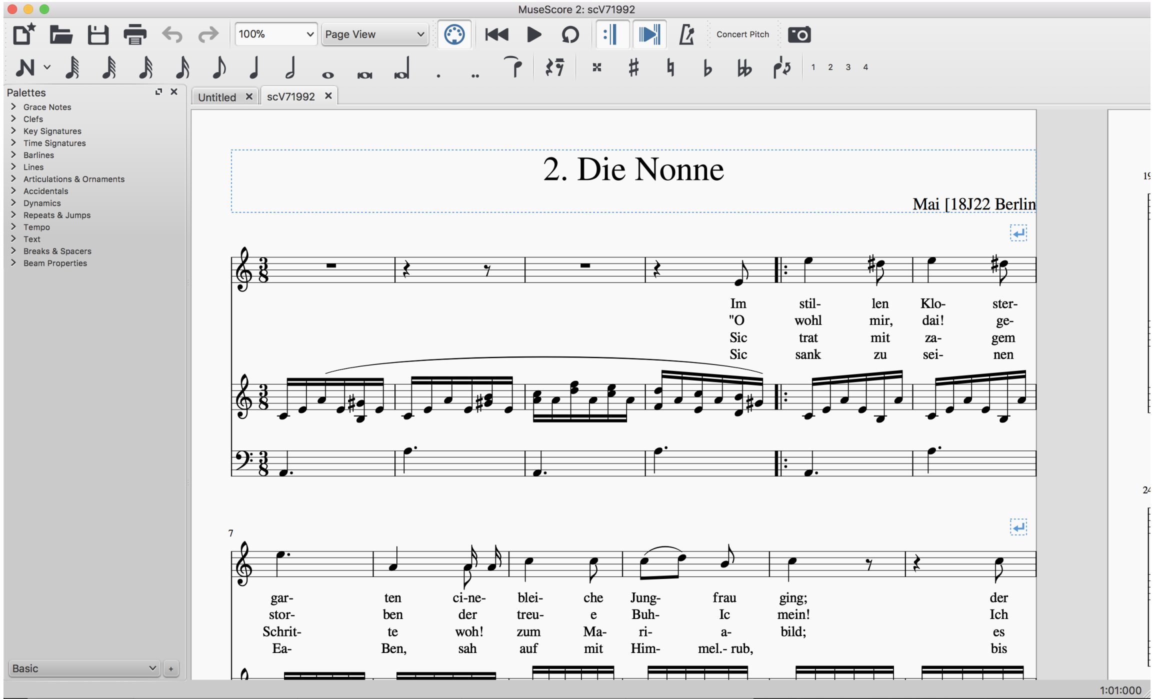

Figure 2.3: A score in MuseScore format after being read by an OMR like PhotoScore or Audiveris

6 minor issues per piece, although there were on occasion major structural issues. Audiveris often added extra beats to measures. It also always identified a bass clef as a baritone clef.

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. One 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 or Audiveris 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: 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). 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. Kern files are organized with columns each representing one staff 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

```
<?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>
```

Figure 2.4: A half middle C in musicXML encoding

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, most lieder consist of 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 kern format by using Humdrum's function `xml2hum` that converts a musicXML file into a kern 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/
```

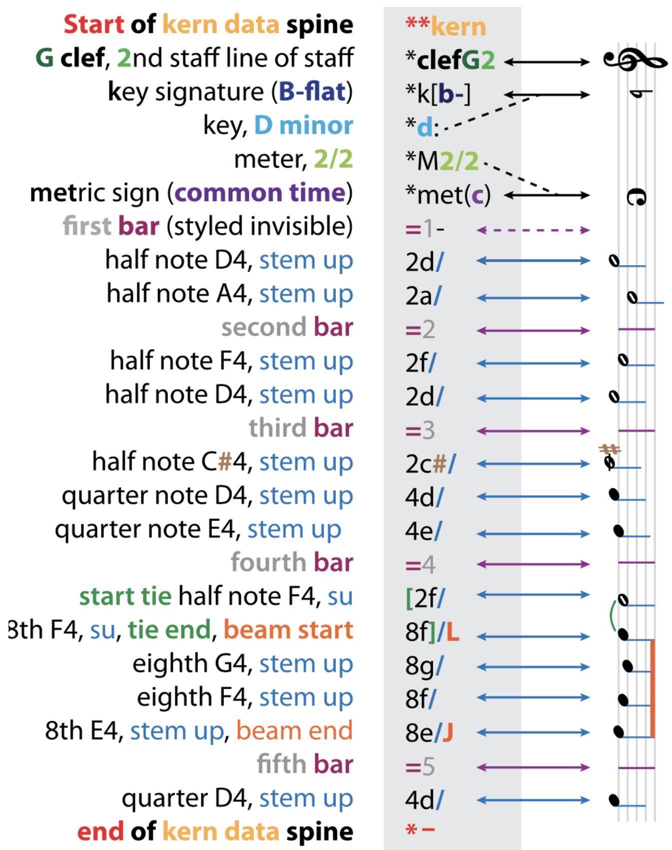

Figure 2.5: How sheet music corresponds to kern files

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 containing work mostly Baroque and Renaissance composers already in the Kern format, which is where the Bach data came from.

The files that were scanned (ie all pieces by Felix and Fanny) need to be separated into separate file for each staff, ie a seperate file for each instrument. In addition, since we are focused on musical style, the text of the pieces is removed in this stage. In the case of analyzing lieder, each piece always has two or three files. These files consist of 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. This is often caused by an inconsistency in time base.

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 rhythem 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 note (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 stem 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 imports sheet music in the proper form (musicXML or Kern) and does all of the analysis using R.

3.1 Importing data into R

MuseR is equipped to import data in the Kern format. The functions for converting these files are `kern2df()` and `piece_df()` These are most usefully in the form of individual splines. This allows for naming the columns according to instrument. Below we have a short piece appearing as it would look from MuseScore. This piece would

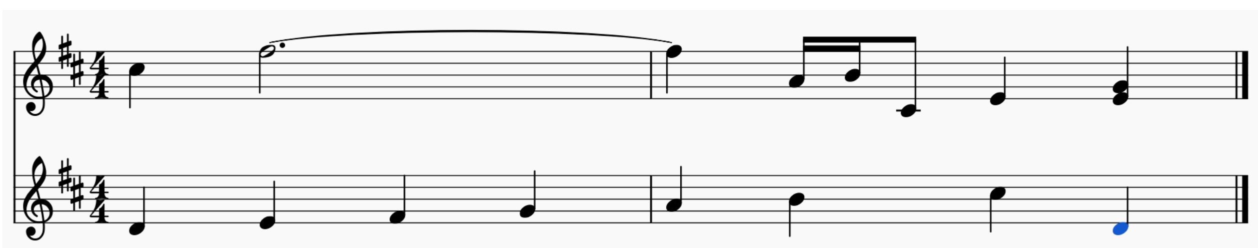

Figure 3.1: Example musescore format excerpt

have the following Kern format representation: Using museR's `piece_df()` function to create the following data frame. Data in R are commonly expressed as data frames. Expressing music as a data frame has challenges, as music cannot be easily expressed in rectangular form. Music as a data structure is very rugged. When expressing it in rectangular form, there needs to be placeholder or padding entries to account for the nonrectangularness. MuseR's `piece_df()` works by using regular expressions to extract note and rhythem information. It uses NA and . values to indicate empty spaces and duration respectively.

The output from `piece_df()` follows the same ideas as the structure of kern files. It has "." values similar to the Kern dots that represent the duration of the note for

```
**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/
==  ==  ==
*-  *-  *-
```

Figure 3.2: Example kern format excerpt of the above musescore

the timebase. This kind structure is good for certain types of features, where we are intersted only in the type of note happening, the time of the attack of the note, or without considering rhythem. Sometimes, we are intersted in how long the note lasts. The `durration_df()` function corrects this issue. It converts the `NA`s representing duration and replaces them with the note value that is currently happening. This allows for analysis that considers duration. The below figure represents the durrational version of the above data frame.

3.2 Features currently supported in museR

Melodic intervals

Melodic intervals, or the interval between two successive notes, are found using the `mel_ints()` function. It is currently only equipped to look at melodic intervals for the top note of each staff. In this context, it is most commonely used for analyzing melodic intervals of the voice. The function first extracts the top line of any instrument, and 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: $\{(f,c),(c,d),(d,f)\} = \{p4, M2, m3\}$,

	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

Figure 3.3: Example R data frame converted from the above MuseScore

	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

Figure 3.4: Example of durrationaly conveted data frame equivalent to the above data frame (need to change)

which would output the proportion vector $(0, 1/3, 1/3, 0, 0, 1/3, 0, 0, 0, 0, 0, 0)$.

If we are interested in the types of melodic intervals, we can use the `connsonance()` to examine the proportion of consonant (perfect, imperfect, dissonant) intervals over the piece. This function works by calling `mel_ints()` and then adding up the perfect, imperfect, and dissonant intervals proportions.

Density

The `beat_density()` function analysis the average and standard devaition of density of each measure in the piece. It is called "beat" density as it only accounts for the instance a note starts. For example if a measure consisted of a single whole note it would be only counted once even though it is voiced the entire measure.

In the above example, the first measure would have a beat density of 4, and the second measure woudl have a beat density of 2.

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 Kern file. Kern files

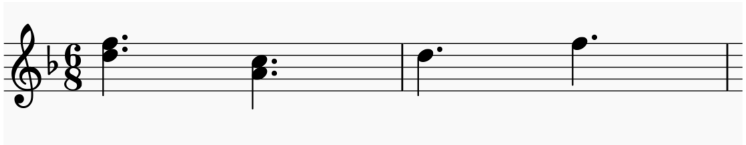

Figure 3.5: Example of calculating proportion of melodic intervals

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 Kern 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.

Prop scale degree

Once the key of a piece is determined, the proportion of each scale degree is calculated. The scale degrees consist of : Tonic, Supertonic, Mediant, Submediant, Dominant, Submediant, and Leading tone. In the example above, if we assume the key is F major, the tonic has a tonic scale degree proportion of 2/6, the mediant 1/6, dominant, 1/6, submediant, 2/6.

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.

Length

We determine a measure for the length of a piece by how many measures the piece has.

3.2.1 Voice distance

The voice distance for each piece is measured as the range of the singer, which is the distance in half steps between the lowest note the singer sings and the hightest.

3.3 Piano distance

Piano distance is the maximum distance between the lowest note in a chord and the highest. Composers with different hand size could possibly have different comfortable chords to play.

Chapter 4

About the Models

Classification problems attempt to divide observations (of features) into gropus (composer) based on similarities in the observations. The notation used in this chapter is inspired by *The Elements of Statistical Learning* (Friedman, Hastie, & Tibshirani, 2001) and *An Introduction to Statistical Learning* (James, Witten, Hastie, & Tibshirani, 2013). Our features space X is an $n \times p$ matrix, where n is the size of our data, and p is the number of predictors. Each X_i is vector of values for a certain feature. x_{ij} denotes the i^{th} values of the j^{th} feature. This X contains the information of the extracted features that hopefully will be helpful in determining the composer of the piece. If the features are different enough between the composer, ie, that the features endode some sense of unconscious (or conscious) use that differes between composer. We can then build and fit models that can use how the features differ between composers. These models can both explain the relationship and difference of the features between composers, and use the use the way the model explains the differences (fitted model) to predict the composer of a piece if we know the same features for that piece.

In addition to the features for each piece, each piece has a composer (response), known or unknown, that we denote by Y. The i^{th} piece has composer Y_i where $i \in 1, \ldots, n$ In our case we have $Y \in \{\text{Fanny}, \text{Felix}, \text{Bach}\}$, or more generally, $y \in \{\text{list of composers}\}$. Since the options for composer are in a discrete set, we can divide the input features space into different groups, or regions, that are labeled according to the classification of composer a model assigns or predicts.

4.0.1 Linear Methods for Classification

4.0.2 Linear Regression:

A naive model for classification is linear regression. We can think of linear regression as plotting all of our points on axes for their feature, and trying to fit a "good" line through the data. If our predictor space Y has K classes, we code the response K different indicator responses y_k where $y_k = 1$ if $Y = k$ 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 $\{x : (\hat{\beta}_{k0} - \hat{l0}) + (\hat{\beta}_k - \hat{\beta}_l)^T x = 0\}$ 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 aren't 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 separately 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 look 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 separated 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 functions 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 separated, 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.

Naive Bayes

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

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

. In practice this is not the case, but the model still preforms surprisingly well in practice when this assumption does not hold.

Logistic Regression

Logistic regression differs 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 likelihood. 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.

K- nearest - neighbor (expand)

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 closeness, 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.

4.0.3 Dimensionality Reduction

Principal component analysis

Principal component analysis (PCA) tranforms the features space into a lower dimensional representation. It choses the transformed features to have maximal variance and be mutually uncorrelated.

Principal component analysis can be useful when the predictors are correlated. We suspect many of our features are correlated, due to certain patterns in music, as well as the way we created our features. These relationships are caused by similarity in the features, and from music theory rules. (For example, if there is a high frequency of first scale degrees, we might expect a high frequency of chords that include the first scale degree. Another example, if we had a high frequency of seventh scale degrees, we would expect them to resolve to the first scale degree.)

Principal component analysis is also helpful when there are many predictors, and we want to deal with a smaller dimension of predictor space.

As an unsupervised method, PCA can inform about latent meta variables.

Used in supervised methods, the transformed features from PCA can be used to fit models instead of the original features space.

Principal components transforms the feature space. If our original features are $X_1, X_2, \ldots, X_p$, we transform the features to $Z_1, Z_2, \ldots, Z_M$, where $M < p$. Each Z_i is a linear combination of the original predictors, ie, $Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$, for constants $\phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$ for $m = 1, \ldots, M$. Given an $n \times p$ data set $\mathbf{X}$ where x_{ij} is the i^{th} instance of the j^{th} feature, we solve for the m^{th} principal component loading vector $\phi_m = \phi_{1m}, \phi_{2m}, \ldots, \phi_{pm}$ that solves the optimization problem:

$$\max_{\phi_{1m}, \ldots, \phi_{pm}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2 \right\}$$

, where the ϕs are subject to $\sum_{j=1}^{p} \phi_{jm}^2 = 1$. Our principal components are then calculated as $z_{im} = \sum_{j=1}^{p} \phi jm x_{ij}$

The loadings of the first principal component, ϕ_1 thus determine the direction in the feature space with the most variance, Z_1, or the scores of the first principal component is then a new feature in our transformed feature space. We continue calculating Z_i, where each following Z_i has the maximal variance in a direction uncorrelated to the previous principal components.

Before PCA is performed, we center all features to have mean zero, as the scale of some features are not the same, which will lead to issues in the loadings, as the features with higher scales would automatically have the higher variance.

We can observe the proportion of variance explained by each principal component. This is usually visualized in a skree plot. We can use this information to decide how many principal components to use.

4.0.4 Model selection/assesment

5-fold Cross validation (expand)

Cross validation involves fitting a model on a training set, and then using the fitted model to predict the responses in the testing set. k-fold cross validation involves splitting the data set into k different parts of equal size, and fitting a model on the the data set with one of the k parts withheld. The model is then tested on the witheld data. This is useful for model selection and determining the accuracy of the model. In this paper, as is common, we use $k = 5$

4.0.5 Lasso model selection

The Lasso penalty was proposed by Robert Tibshirani in 1996. Lasso regression works by giving a penalty to regression coefficients. It essentially performes variable selection, as for high enough penalties, coeficcients shrink to zero. It is often used in linear regression, but can be expended to logistic regression and other generalized linear models. For linear regression, the lasso works by choosing coefficients β_λ^L that minimize

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

We have that λ is a tuning parameter. Increasing λ will shrink the coefficients. This can be equivantely stated as:

$$\text{minimize}_\beta\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \text{ subject to } \sum_{j=1}^{p}|\beta_j| \leq s\right.$$

To expand to generalized linear models, if our model uses some parameter β that was estimated by by some function $\ell(\beta)$ (where $\ell(\beta)$ is a log-likelyhood function for example), we maximize $\ell(\beta)$ subject to $\sum_{j=1}^{p}|\beta_j| < s$ (Tibshirani, 1996)

4.0.6 Some trees?

4.0.7 K-Measns Clustering (plus other clusterings?)

Chapter 5

Exploratory Data Analysis

5.1 Bach and Mendelssohns

Below (will be?) the pairwise distributions of each of the features used for Bach and the Mendelssohn s. There are higher correlations between scale degree frequencies 3 and 6. There is also a surprising separation between the composers for the mean density. The above biplot shows the loading vectors plotted on the first two principal

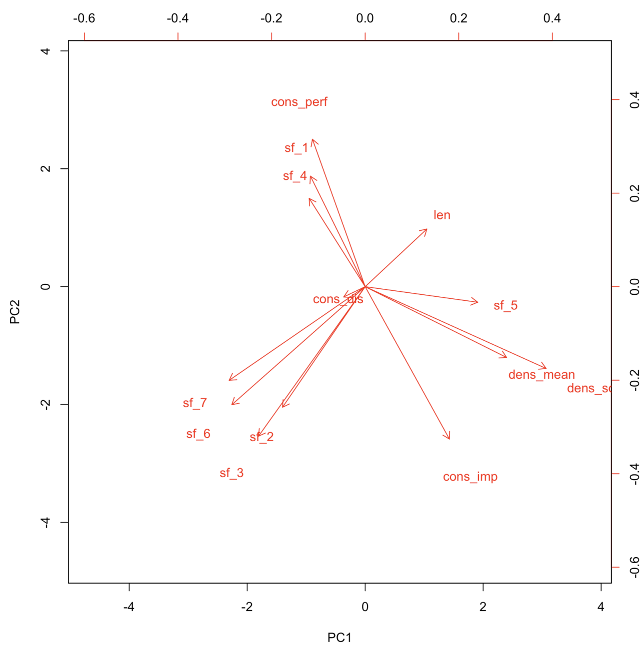

Figure 5.1: Biplot of the loading vectors of the first two principal components

components. The loading vectors of features seem to arrange into aproximately three groups. The features: perfect consonances of melodic intervals of the voice, frequency of the first scale degree, frequency of the fourth scale degree are all grouped together. Similarly, frequencies for the 7th, 6th, 2nd, and 3rd are grouped together. In addition we have perfect consonances and imperfect consonances on two sides of the second principal component. This leads me to believe that the second principal component encodes some sense of use of consonant notes. The first principal does not seem to have a musical interpretation. (I think?) The above? plot shows the same prinicpal

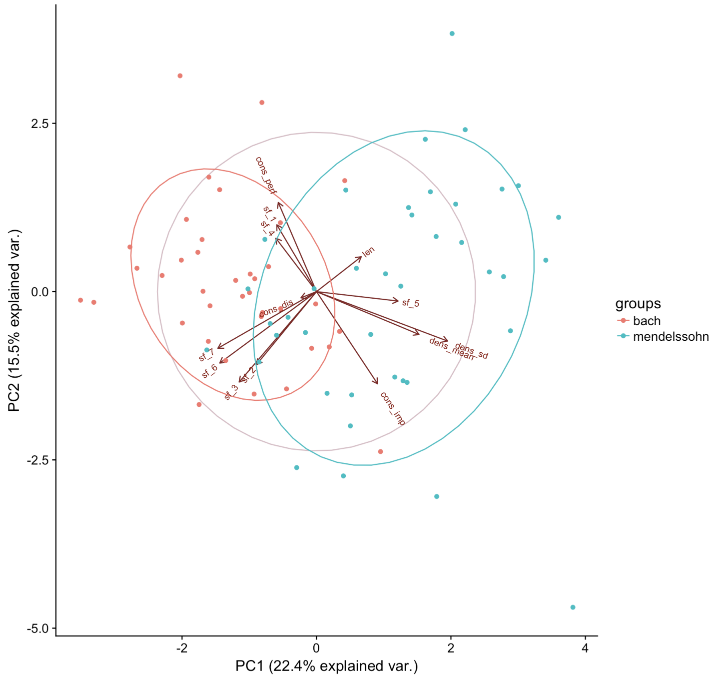

Figure 5.2: Biplot of the first two principal components plotted with data points colored by composer.

components, with the addition of points representing each piece graphed by their first two principal components. The elipses represent a ..... The pieces are colored by thier composer. There does seem to be a decent amount of seperation between the groups, although it is not complete. The below plot shows the pieces plotted on the second and third principal compenent, and there is much less seperation. Also there does not seem to be an obvious musical interpretation of the loadings. The skree plot shows no clear elbow. Because of this, and also because we do not have too many predictors, we will choose to use $p - 1$ principal components in subsequent analysis, where p is the number of predictors in our feature space.

We can also use K-means clustering to see if there are any apparent groups in the data. When we run K-means on the bach and mendelssohn with K = 2 (since there are two composers) We can also examine when $K = 3$, as the Mendelssohn set is made up of fanny and felix, so there are actually three composers. (fix labeling)

5.2 Felix and Fanny

There will be pairwise scatterplots of features? Which ones? Also distributions of composer for certain features? Which ones? Who knows.

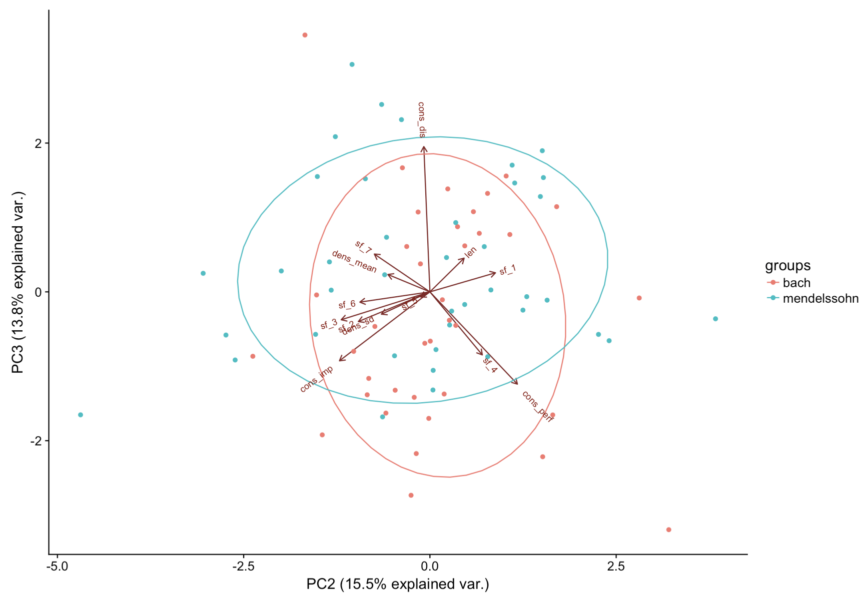

Figure 5.3: Biplot of the second and third principal components plotted with data points colored by composer.

We can see from the above pairwise distribution plot that we still have a high correlation between scale degrees 3 and 6. However we do not see the difference in beat density. This is likely due to the fact that fanny and Felix data set is composed of the same type of song, whereas the Bach data set is only solo piano music. There doesn't seem to be as much of a musical interpretation of the loadings of the principal components. We do notice that the frequency of perfect consonant melodic intervals in the voice (fcons_perf) has an almost opposite signs of the second principal component to the frequency of imperfect and dissonant melodic intervals. The skree plot also doesnt seem to show a clear elbow as indication of how many PCs to choose in subsequent analysis. Unfortunately, as we see in the above picutre, there is absolutely no seperation in the groups. This might lead to issues in creating a model to classify. When we run K-means with $K = 2$ we get the following. It does not seem choose groups by composer.

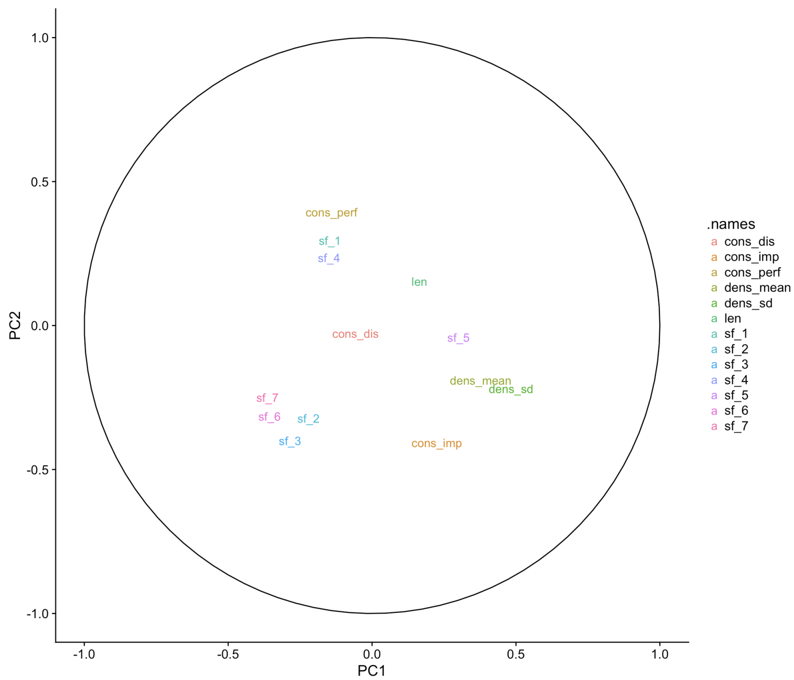

Figure 5.4: Loadings plotted on a unit circle

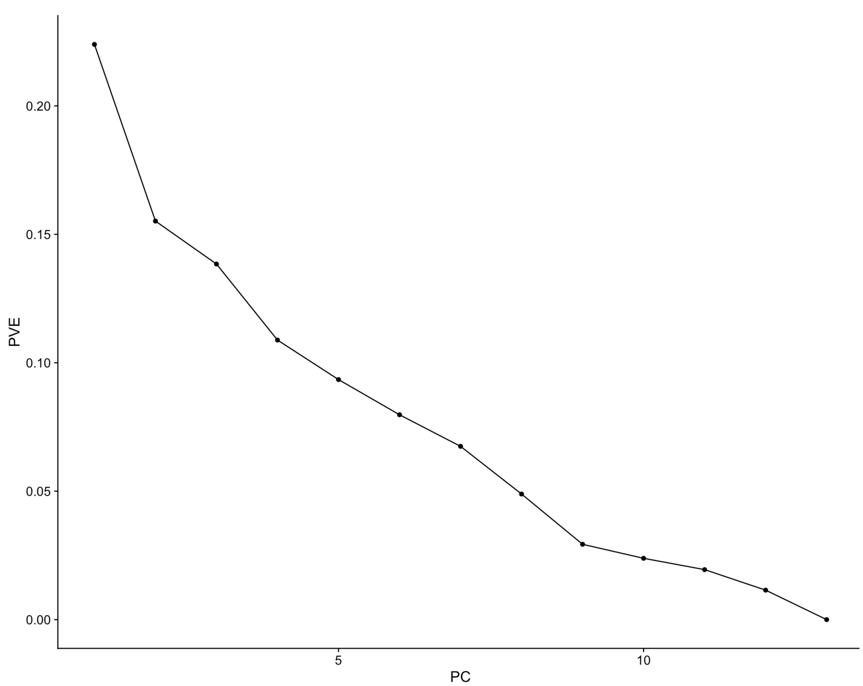

Figure 5.5: Skree plot of the PCA's

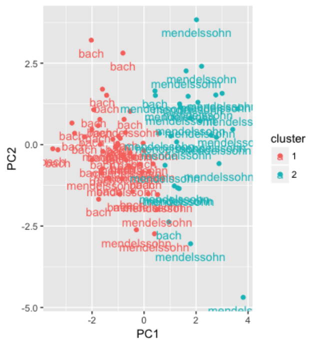

Figure 5.6: KNN when k = 2

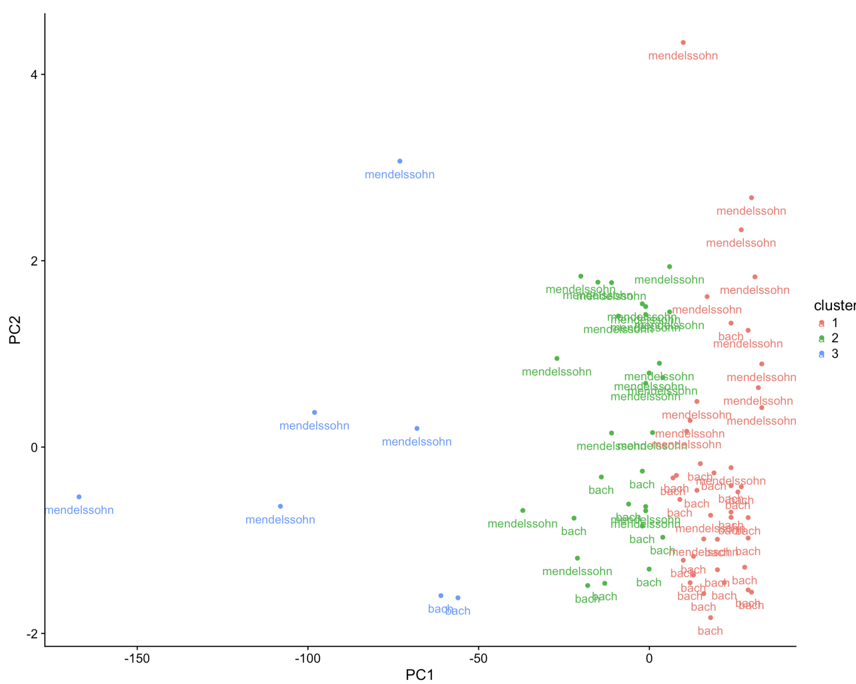

Figure 5.7: KNN when k = 3

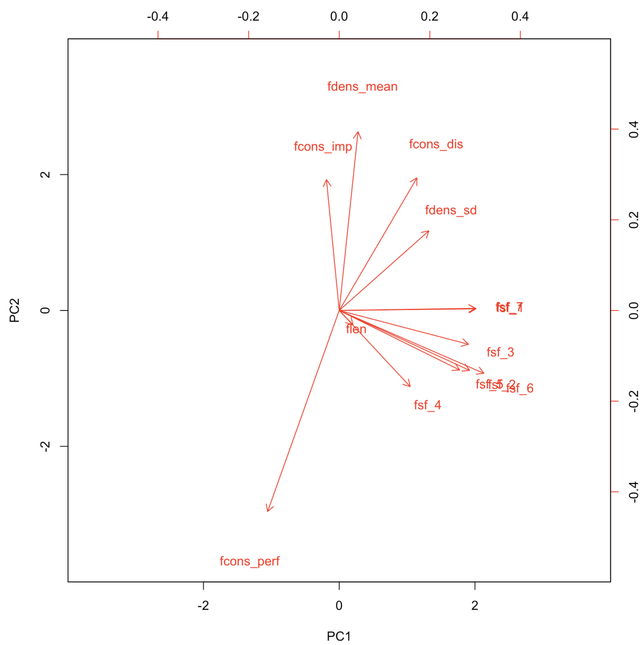

Figure 5.8: PCA Loadings of the features for Felix and Fanny

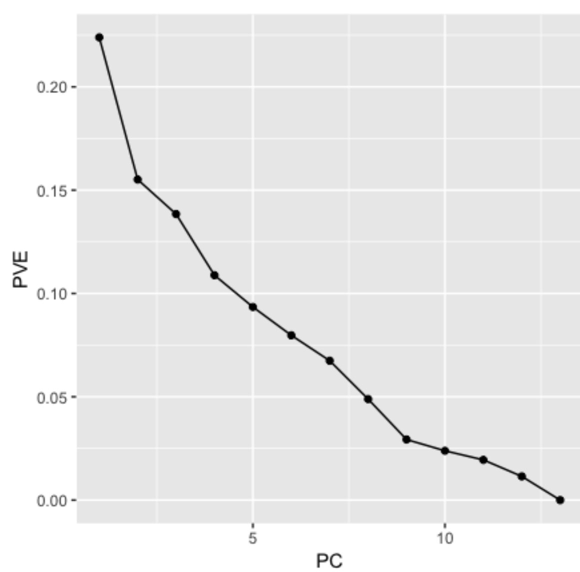

Figure 5.9: PCA felix/fanny skree plot

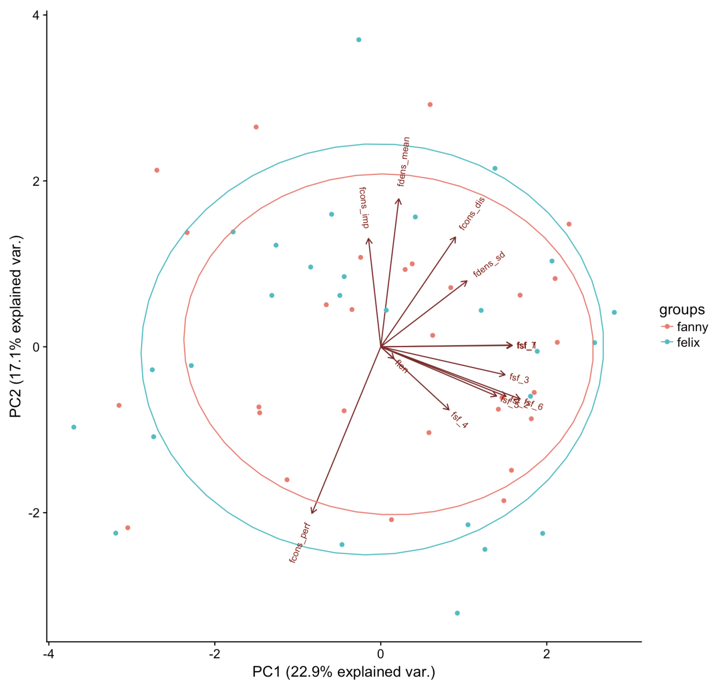

Figure 5.10: Loadings and pieces plotted on the first principal components.

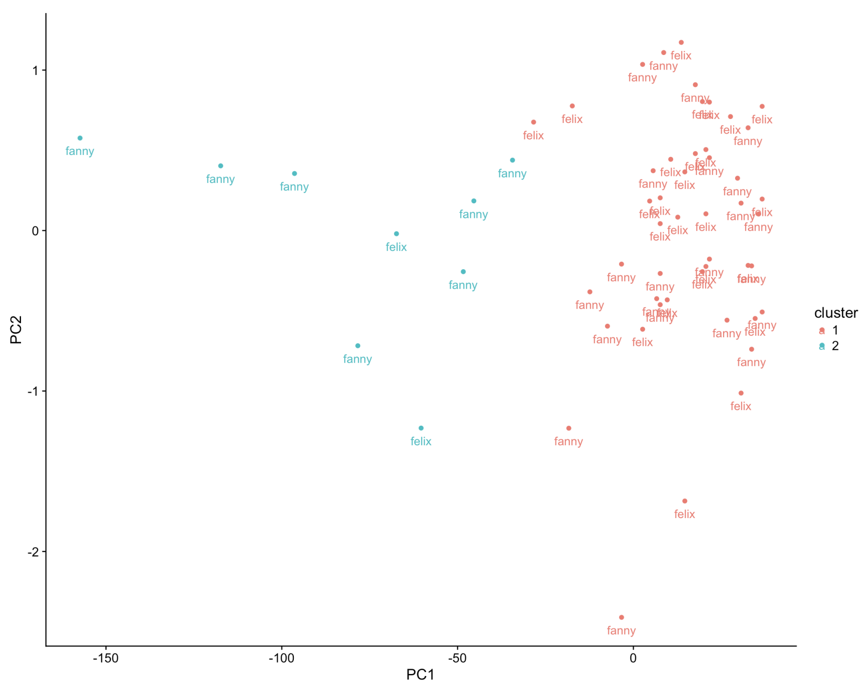

Figure 5.11: K-means with k = 2

Chapter 6

Model Fit

6.1 Model Fit - Bach/Mendelssohn

Logistic Regression

A 5-fold cross-validated lasso logistic model was fit. The below graph shows the miss-classification rate for different values of log(lambda). A model fit using the lambda value with the minimum missclassifcation rate resulted in a miss-classification rate of 0.117. We see that the density features stayed in the longest.

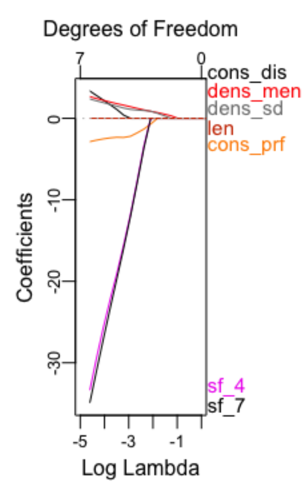

Figure 6.1: Lasso penalties for each feature for changing lambda penalty values

LDA

For linear discriminant analysis, we have a 5-fold cross validated MSE of 0.13.Most commonly our models incorrectly predict Mendelssohn songs to be composed by Bach.

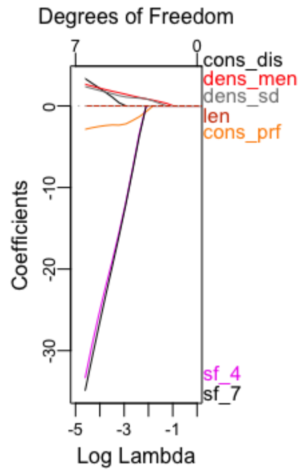

Figure 6.2: Cross validated misclassification rates for different lambdas. The left dotted line represents the minimum lambda, and the right line represents the lambda within one standard deviation. It is a more restricted model that can guard against over fitting, but is not used here.

Naive Bayes

A naive model was fit and resulted a miss-classification rate of 0.07

KNN

A K-Nearest Neighbors classifier was fit, with K chosen to be 1 from 5-fold cross validation. This resulted in a miss-classification rate of .22

6.2 Model fit Felix/Fanny

It there seems like there is not enough data to have a stable missclassification rate for different choices of fold.

Logistic regression

When a logistic classifier was fit, we got an average miss-classification rate of 0.35. We see that the density features and fifth scale degree stayed in the longest.

Naive Bayes

A naive Bayes model was fit and the 5-fold miss-classification rate averaged between .4 and .5.

LDA

An LDA model was fit and predicts correctly about 50/50

knn

With K=8, the missclassification rate was about .43

6.3 Predictions for Disputed pieces:

Since we have such high missclassification rates, the following predictions are likely not accurate.

- Logistic: Predicts Op 8 no 2 and 12 to be written by Felix.
- Naive Bayes: Predicted all four as written by Fanny.
- knn: Predicted Op 8 no 12 to be written by Felix.
- LDA: Predicted Op9 no 12 to be written by Felix.

Chapter 7

Discussion

7.1 Discussion

On very basic low-level features, consiting mostly of frequencies of notes, intervals and chords, most models comparing Bach to the Mendelssohns do relatively well. This is likely due to the decent seperation of the feature encoding density as shown below. This is likely because the Bach data is for solo piano and the Mendelssohn data has an additional instrument, thus makeing the piece automatically more dense.

Table 7.1: Miss-classification Rates for differnt models comparing Bach and Mendelssohns

Model	Miss-classification Rate
Naive Bayes	0
KNN	0
LDA	0
Logistic	0

On the other hand, models fit to compare Felix and Fanny did not do as well. They are only very slightly better than random guessing.

Table 7.2: Miss-classification Rates for differnt models comparing Felix and Fanny

Model	Miss-classification Rate
Naive Bayes	0
KNN	0
LDA	0
Logistic	0

7.2 Future suggestions for museR

Future suggestions to improve functionality of the R package museR include equipping it to deal with higher level features…

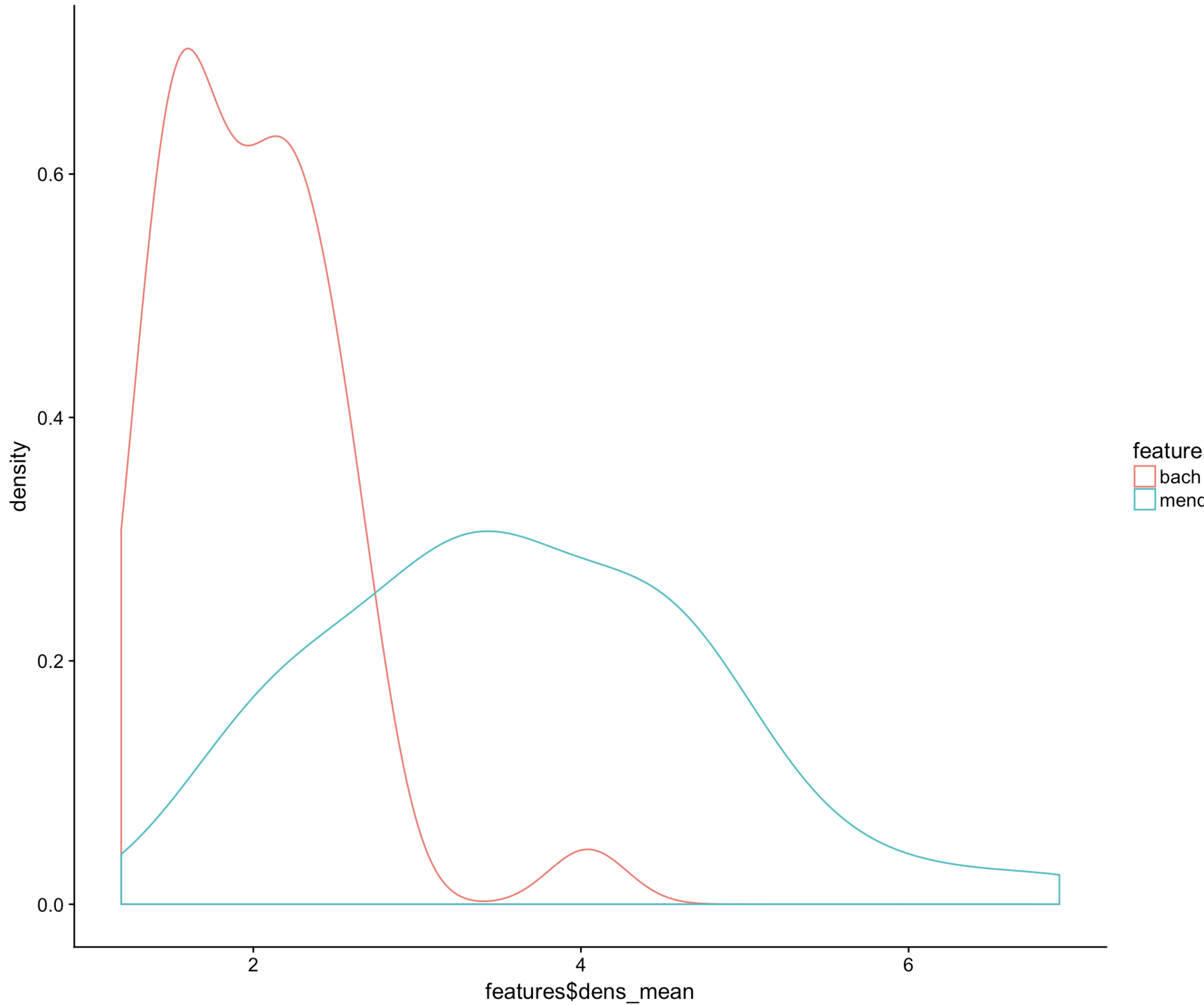

Figure 7.1: Distribution of feature of the mean beat density for Bach and Mendelssohn

Conclusion

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

```
# 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)
library(MASS)
library(class)
library(tidyverse)
library(tree)
library(randomForest)
library(e1071)
library(ggplot2)
library(GGally)
library(ISLR)
library(boot)
library(glmnet)
library(caret)
library(plotmo)
library(museR)
```

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.

Ayrton, W. (1830). *The harmonicon* (Vol. 8). W. Pinnock.

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`

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer series in statistics New York.

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

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

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/`

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

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

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