Analyzing Sheet Music

Eric Hurst, Kyle Ray, Chase Wilson, and Tobin Yehle

Abstract—This study examines the use of machine learning methods to read printed sheet music and classify note duration and pitch. Applying Gaussian Discriminant Analysis (GDA) on features extracted from sheet music, classification of notes duration and pitch are learned on a training set and applied to a testing set. This approach results in relatively successful classification of notes, but does not perform particularly well when classifying pitch.

I. INTRODUCTION

Sheet music is a handwritten, printed, or digital form of music notation using modern musical symbols. For the purposes of our study, digital sheet music (in .pdf or .jpg format) is the medium of choice. Modern music notation uses a five-line staff. Pitch is shown by placement of notes on the staff. A musical note is a sign used in musical notation to represent the relative duration and pitch of a sound.

The field of sheet music analysis using machine learning techniques is a promising one. A musician could scan a tune and have it transposed automatically, a score could be translated to Braille with little effort, or corrections could be made to an outdated edition by comparing the new edition with the outdated one.

Our objective is to classify notes by duration and pitch using methods of machine learning, specifically a GDA classifier. For our study, the goal is to classify note duration correctly as either quarter, half, or whole note and classify pitch under the assumption that the treble clef indicates the value of the pitches on the staff.

We approach our problem of finding note duration first by using a horizontal summation of whitespace technique and K-Means clustering algorithm on a sheet of music to remove any text on the page and separate the full sheet of music into individual 5-line staff images, this step is then repeated in order to isolate each individual note.

Each note duration is then classified using a GDA classification algorithm with total brightness as the sole feature. We use the brightness measurement to find the vertical position of the geometric centroid of darkness in the data. The vertical position of the centroid is used to classify note pitch.

This approach is appealing since it makes for basic feature vectors. Since digital sheet music is simply black or white, we predict that using overall brightness in our feature vector will work well.

The results of our study indicate that the note duration classifier is far more successful in classifying quarter, half, and whole notes (at 85% accuracy) than our pitch classifier (at 33% accuracy). Though, the accuracy of our pitch classifier is not as bad as it seems since there are 15 possible pitch classes. This means our pitch classifier, in fact, performs for better than simply guessing the pitch.

II. RELATED WORKS

Research into printed sheet music recognition was performed at the graduate level at MIT and other institutions in the late 1960s using Music Optical Character Recognition (Music OCR). Their technique was to remove the staff lines from the image data, and then try to recognize the symbols that remained. There have been commercial music-scanning products since the early 1990s when MIDISCAN was released.

There are many products available today that can digitize sheet music, but they all have enough problems that most musicians do not use them. There are many commercially available software suites that include the ability to scan sheet music, but since they are part of a larger software package any scanning features remain an underused and underdeveloped piece of functionality. A few open source solutions exist, but they also have problems. Software such as Audiveris [2] uses complex time consuming machine learning algorithms such as neural networks to solve the problem of non-standard input which leads to low accuracy. The low accuracy of existing solutions means sheet music scanning technology remains largely ignored by the music community.

Another field of related works is known as music information retrieval (MIR). While our goal is to classify sheet music (without the use of audio data), MIR primarily concentrates on audio data to classify music into genres or to transcribe an audio recording into symbolic notation. It appears as though the primary limitation of MIR (with respect to reading sheet music) is the lack of access to volumes of sheet music. In our study, we find this accessibility issue (specifically, lack of accessibility to data with labels) to be a definitive limiting factor.

While these works are related, there is no evidence to indicate that analyzing sheet music using simple features and Gaussian Discriminant Analysis for classification has been attempted before.
III. PRELIMINARIES & PROBLEM DEFINITION

A. Scope

The number of symbols in common sheet music is very high, which is part of the reason doing character recognition is hard. In our solution we will deal with a smaller subset of sheet music. Our music will feature whole, half, and quarter notes that can be placed anywhere on the staff lines without ledger lines. Ties, accents, staccato markings, and other annotations to notes will not be present. We will focus on classification of notes, rests, and other symbols such as clefs. We will ignore tempo and markings as well as key and time signatures. This will make note extraction more feasible and allow classification to be effective on a smaller data set. An example of the type of music we will use as data can be found in the appendix.

B. Assumptions

There are several assumptions our process makes about the state of the input data. We assume the image is not slanted. This means our method of finding staff lines using summation of whitespace will work well. The other assumption we make is that the note and rest symbols are similar to the ones in our training set. There are several styles of writing rests, and we will only train on the style used in modern compositions. Another assumption of Gaussian Discriminant Analysis is that the relative frequency of classes is the same in the training and testing data. This means our training data should have very few clefs compared to the number of notes. This assumption will be fulfilled easily in our tests because of the way we choose data to test on, but in a real application it may be useful to train on a specific set of data you know will be similar to the page you are scanning. This is possible only if you use a classification algorithm with a simple training step such as Gaussian Discriminant Analysis.

C. Gaussian Discriminant Analysis

Gaussian Discriminant Analysis (GDA) is a supervised learning algorithm that takes labeled data and returns a classifier for each label. The algorithm assumes the data in each class is distributed according to a Gaussian distribution. It fits a Gaussian distribution to each class and assigns each class a weight. The weight is a reflection of the frequency of each class in the training data. This means the data we train on must have a similar frequency of symbols as we are expecting in our testing data. To classify a new data point the classifier finds the Gaussian distribution most likely to produce the new data point.

Finding the parameters for GDA only requires a single maximization step. The complexity of this step depends only on the number of data points in the training set, and the dimensionality of the feature vector. GDA requires data, \( x, y \), and a number of classes, \( m \), where \( x \in \mathbb{R}^n \) and \( y \in \{0 \ldots m\} \). The parameters for the model are \( \phi, \mu, \) and \( \Sigma \), where \( \phi \) is a list containing the probabilities of each class occurring (the weight of the class), \( \mu \) is a list of vectors describing the mean of each class, and \( \Sigma \) is a list of matrices describing the variance of each class. We start with the following information

\[
y \sim \text{Bernoulli}(\phi)
\]

\[
x | y = j \sim \mathcal{N}(\mu_j, \Sigma_j)
\]

The likelihood of the parameters producing the data is

\[
\ell(\phi, \mu, \Sigma) = \prod_{i=1}^{n} P(x_i | y_i) P(y_i)
\]

\[
= \prod_{i=1}^{n} P(x_i | y_i) \prod_{i=1}^{n} P(y_i)
\]

\[
= \prod_{j=1}^{m} \prod_{i=1}^{n} P(x_i | y_i = j) \{y = j\} \prod_{i=1}^{n} P(y_i)
\]

\[
\ell = \sum_{i,j=1}^{n,m} \{y = j\} \log P(x_i | y_i = j) + \sum_{i=1}^{n} \log P(y_i)
\]

Maximizing this likelihood function for each of the parameters of the model gives the following equations for each of \( \phi, \mu, \) and \( \Sigma \):

\[
\phi_j = \frac{1}{n} \sum_{i=1}^{n} 1 \{y_i = j\}
\]

\[
\mu_j = \frac{\sum_{i=1}^{n} 1 \{y_i = j\} x_i}{\sum_{i=1}^{n} 1 \{y_i = j\}}
\]

\[
\Sigma_j = \frac{\sum_{i=1}^{n} 1 \{y_i = j\} (x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{n} 1 \{y_i = j\}}
\]

In order to classify a new data point we need to find the probability of each label given the new data point.

\[
P(y = j | x) = \frac{P(x | y = j) P(y = j)}{P(x)}
\]

\[
= \frac{\sum_{k=1}^{m} P(x | y = k) P(y = j)}{\sum_{k=1}^{m} P(x | y = k) P(y = k)}
\]

Since the only piece of information we care about is which probability is the greatest we can ignore the denominator. The classifier is then given by the following equation.

\[
y = \max_{j=1}^{m} (P(x | y = j) P(y = j))
\]
To test how well the classifier works we train it on a portion of the data and test it on the remaining labeled data points. The percentage of the remaining data points the classifier labels correctly is the accuracy of the classifier.

IV. APPROACH

We approach our solution in two phases. First, we must extract the individual note and other symbols from the page of sheet music. This involves pre-processing the image using basic image processing techniques. Once we find the individual notes and symbols, we must then classify each symbol as a note, or other symbol. In addition, if the symbol is a note, we continue to find both the pitch and length of the note. Similarly, if we find a rest, we also try to predict the length of the note.

A. Note Extraction

In order to extract symbols from a page of sheet music, we start by identifying the most distinguish element of the page, the staff lines. After we extract staff lines from the page, we can then use a similar method to extract notes from each staff line.

In order to extract both staffs from a page, and notes from a staff, we will use a method called horizontal summation of whitespace. We first assume that our image is thresholded so that there are only two values within each image. 0 will represent a black pixel, and 1 will represent a white pixel. We then take the summations of pixels for each row in our image. Assuming \( I \) is our image, and \( I_{ij} \) is the pixel in the \( i^{th} \) row and \( j^{th} \) column, then we can represent the horizontal summation of whitespace approach as

\[
S_i = \sum_{j=1}^{i} I_{ij}
\]

where \( S \) is a vector containing the horizontal whitespace summations for each row. A small example of this method can be seen in Fig. 1.

In Fig. 2, we can see an example of such a vector plotted. By visual inspection, we can see three big features, or groups of concentrated blackspace. These correspond the three staff lines that are present in this sample image. We can now take a threshold of this image, and group staff lines based on contiguous segments of blackspace below a certain threshold.

A similar method is used for extracting notes from each staff line. By rotating each staff line by 90°, we are able to use the horizontal summation of whitespace on each staff line to find symbols on each staff line.

B. Note Recognition

Once we have each note or symbol extracted from the page of sheet music, we then classify the symbol’s type, and possibly pitch and length.

For type classification, we use a simple Gaussian Discriminant Analysis in order to classify each different symbol. Based on our training data, we look for a number of types of symbols. Various symbols that we looked for were notes, rests, treble clefs, measure bars, and time signatures. There are many other symbols that may be found on a page a sheet music, and this method can easily be extended to classify these additional symbols. In order to help the GDA to distinguish between different types of symbols, we used the average brightness of the extracted images to classify different symbols.

Pitch classification uses the same Gaussian Discriminant Analysis to classify different pitches based on the learned data. However, pitch classification is more challenging since each note’s pitch is dependent on relative position to the staff lines. Since pitch is determined how high or low a note is on the staff, the main feature that we utilize is the vertical centroid of the image. A histogram displaying learned parameters of our note pitch classifier using GDA can be seen in Fig. 3.
Similarly, a histogram displaying learned parameters of our note duration classifier using GDA can be seen in Fig. 4.

V. RESULTS

A note is generally comprised of two attributes: a type (or duration) and a pitch. The type determines if the length of the note and if it is a note, a Clef, or a rest. The pitch is the sound it makes and is determined by its location on the five staff lines. To classify the note, we broke the classification in two parts. The first part will classify its type and the second part will classify its pitch.

A. Type Classification

To classify its type, we took the average brightness, amount of whitespace, at each line of the note image and then put 90% through the GDA classifier with 5 classes (Whole Note, Half Note, Quarter Note, Clef, and Rest). Then putting the other 10% through the trained classifier to test accuracy yielded a 85% accuracy rate.

B. Pitch Classification

Pitch classification is a bit more complex since it there are 15 different classes to classify. We took the vertical position of the geometric centroid of the image which should have been the position of the note on the staff lines. We then used 90% of the data to put in the GDA classification to get the classifier for each of the pitch. Then using the rest of the data to test the accuracy of the classifiers, we got a 33% accuracy.

While this may seem bad at first, it is a lot better than random guessing which would be 6.67%. The main reason for this low of an accuracy was because the the note image wasn’t normalized and the staff lines could be at any height on the image heavily skewing the results. By normalizing the staff lines, we believe this will greatly improve the accuracy of the classifier.

VI. CONCLUSIONS

Gaussian Discriminant Analysis performs well when classifying note duration and taking average brightness data as a feature vector. When applied to classifying pitch, on the other hand, GDA performs moderately well (considering there are 15 pitch classes), but not as well as we expected or hoped.

There are a few very apparent aspects of this project that could improve future work in sheet music analysis using GDA. First, the amount of (labeled) data has been limited to sheet music that we have had the time to label manually. Second, extracting additional features to be used in our GDA algorithm should result in more accurate classification. Finally, improving our method to extract notes and staffs would result in much more accurate pitch classification. If the note extraction method could be improved upon so that staff lines in each extracted note would be at the same position across all extracted note data, the vertical position of centroid (and other additional features referred to above) would be much more reliable in its pitch classification.
REFERENCES


APPENDIX

ABC Song

F Bb F C7 F C7 F
A - B - C - D - E - F - G,
H - I - J - K - L - M - N - O - P.

C7 F C7 F C7 F
Q - R - S and T - U - V,
W - X and Y and Z.

F Bb F C7 F C7 F
Now I know my A - B - Cs,
Next time won't you sing with me?

Kiboomu Music 2013
All Rights Reserved