Testing K Nearest Neighbor Classifier with

Fisher Linear Discriminant

CS - 559 | Stevens Institute of Technology

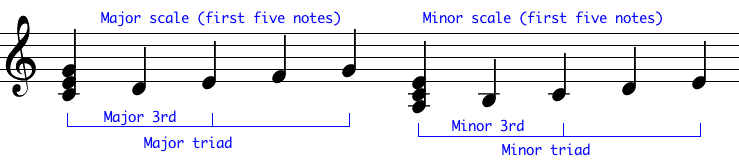
By: Ishaan Patel & Dylan Lopez

May 15, 2021

# **Problem**

The goal of this project is to compare the accuracy of a K Nearest Neighbor Classifier before and after applying FLD. By choosing the “mode” feature as the class, our question is how do we utilize the label data to find accurate projections.

The formal definition of a song’s Mode is - “Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.” The difference between a major and minor chord comes down to one, simple change: the 3rd in a scale. A major chord contains the 1st, 3rd, and 5th degree of the major scale. A minor chord contains the 1st, flattened 3rd, and 5th degree of the major scale of that note.

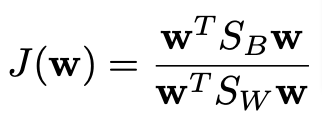


One key takeaway of a musical mode is that the major key is judged as happy-related songs while minor key music is heard as sad, melancholy music.

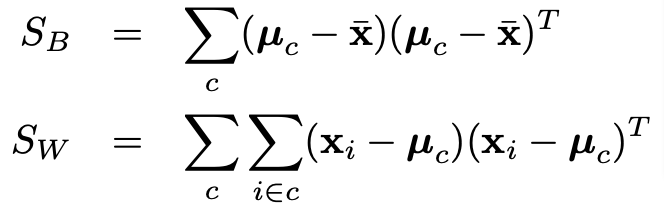
# **Connection to Class Material**

Fishers linear discriminant was used to reduce the amount of features before applying a classifier.

Purpose of Fisher Linear Discriminant is to maximize this function:



SB represents the “between class matrix” and SW represents the within class matrix.



Since the mode class is binary, we were able to find the optimal line direction to classify the data accordingly. From there we projected all data onto one dimension for the training and test data.

# **Train/Test Splits**

Separating the data plays a significant role in model performance with the goal of minimizing any potential discrepancies to gain a better understanding of the model’s results. We randomly split training and test data first within the original dimensions. Then applied KNN onto the dimensions. Then applied FLD on the dataset to reduce it down to a one dimensional space, then ran KNN on the reduced dimensional space to compare model accuracy. We ran it for 10 total trials.

# **Classification**

For this project we chose the K-Nearest Neighbors (KNN) supervised learning algorithm because our data is labeled with the target variable being the “Mode” of the song either 0 or 1. The KNN classifier took each test sample and used the k-closest neighbors from the training dataset to determine the test sample classes. KNN’s are one of the most applied learning models due to its simplicity. Our execution of KNN was dependent on structured data that must be in a tabular structure. We also needed to make sure that the features consisted of numerical data.

# **Dataset**

We selected a dataset provided from Spotify towards a Kaggle challenge. The original dataset consists of over 175,000 songs from years 1920-2020. Due to the large size, we randomly selected 1700 rows each for the training and test sets.

Songs range from all music styles, genre’s and popularity levels. The class was set to “Mode” (column 11). We chose the following numerical features: accousticness, danceability, duration\_ms, energy, instrumentalness, liveness, loudness, speechiness, tempo, valence, popularity, key and mode.

# **Results**

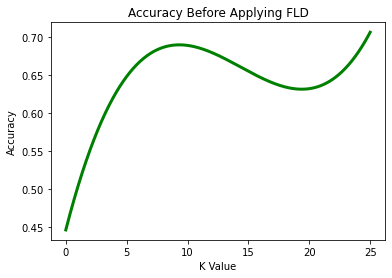
The results remained consistent from before and after applying Fishers Linear Discriminant. By reducing the number of dimensions from 11 to 1, the accuracy did not fluctuate.

We realized KNN’s main disadvantage was that it performed slow on a large dataset, which can make it an inefficient algorithm where implements need to be executed faster. With additional computing resources, we would have been able to effectively run this algorithm on the entire dataset.

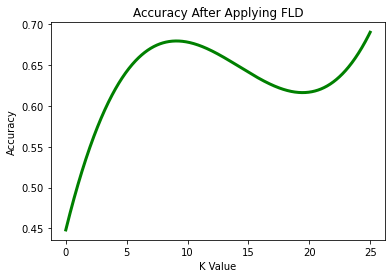
|  |  |
| --- | --- |
| Results for K Nearest Neighbour Classifier for original dimensions for k=5  Accuracy for trial 1: 84.85%  Accuracy for trial 2: 87.32%  Accuracy for trial 3: 16.58%  Accuracy for trial 4: 15.09%  Accuracy for trial 5: 16.87%  Accuracy for trial 6: 83.88%  Accuracy for trial 7: 82.21%  Accuracy for trial 8: 81.24%  Accuracy for trial 9: 84.28%  Accuracy for trial 10: 85.77%  Mean: 0.6380952380952382  Standard Deviation: 0.31225199899149203 | Results for K Nearest Neighbour Classifier after applying FLD for k=5  Accuracy for trial 1: 84.34%  Accuracy for trial 2: 85.77%  Accuracy for trial 3: 18.36%  Accuracy for trial 4: 16.06%  Accuracy for trial 5: 21.0%  Accuracy for trial 6: 82.62%  Accuracy for trial 7: 82.27%  Accuracy for trial 8: 82.39%  Accuracy for trial 9: 81.53%  Accuracy for trial 10: 82.73%  Mean: 0.6370625358577165  Standard Deviation: 0.2965354764024988  Difference between means: 0.00103270223752161 |

When we decrease the value of K under K=5, our accuracy is low (under 60%) and leads to unstable decision boundaries. In addition, as we increase the amount of k neighbors, our accuracy becomes more accurate up to a certain threshold. Eventually, we notice if we increase k too high, it will result in an increase in number of errors. In terms of optimal accuracy, the K value of 25 proved the highest score 78.8% before applying FLD (plot 1.1) and at K=5 we saw the highest value of 76.2% after applying FLD (plot 1.2). The difference in accuracy between before and after applying FTD continued to increase (plot 1.3) Note: The graphs were smoothed with a spline fitting weight of 3 to work around any outliers and discrepancies as the K value increased.

1.1



1.2



1.3

