**Portfolio Milestone: Dynamic Time Warping App**

Chioma Chance

Colorado State University Global

CSC506: Design and Analysis of Algorithms

Dr. Banerjee

03/22/2025

**Portfolio Milestone: Analysis and Optimization**

**Introduction**

This project explores the use of Dynamic Time Warping (DTW) to compare audio files using MFCC (Mel-Frequency Cepstral Coefficients) features. The goal is to identify similarities between audio clips, which can be a foundation for music recognition, sample detection, or even audio-based diagnostics. In this milestone, I analyzed DTW performance across five test files and assessed the time complexity and space utilization, ultimately identifying opportunities for optimization.

**Program Purpose**

The DTW-based app was developed to determine the similarity between two audio files. Specifically, I tested the comparison between samples and full songs using DTW distance. For this round of testing, I used the following test cases:

* test1.wav is a sample of test2.wav
* test4.wav is a sample of test3.wav
* test\_audio.wav is a randomly selected track

These comparisons aimed to determine whether DTW could accurately measure the similarity between sample and original audio clips.

**Empirical Results**

The table below summarizes the output:

| **Comparison** | **DTW Distance** | **Time Taken (s)** |
| --- | --- | --- |
| test1.wav vs. test2.wav | 1,436,533.21 | 310.75 |
| test3.wav vs. test4.wav | 1,189,503.85 | 241.18 |
| test1.wav vs. test3.wav | 1,030,614.22 | 187.02 |
| test2.wav vs. test4.wav | 1,289,277.21 | 67.50 |
| test\_audio.wav vs. test1.wav | 925,171.70 | 62.47 |

Interestingly, the lowest DTW score was not between the true sample matches. Instead, test\_audio.wav vs. test1.wav showed the closest match, which suggests that either the MFCC features or the DTW alignment needs refinement. This indicates a major obstacle in feature extraction or distance normalization.

**Challenges and Limitations**

One of the major issues was performance. DTW has a time complexity of O(N\*M), where N and M are the lengths of the sequences. This resulted in run times over 5 minutes in some cases. Additionally, true sample pairs like test1.wav vs test2.wav did not yield the lowest distances, showing that the current setup does not adequately reflect sample similarity.

**Optimizations and Improvements**

1. **FastDTW**: A linear approximation of DTW could significantly improve performance while maintaining decent accuracy.
2. **Feature Reduction**: Reducing the number of MFCC coefficients or downsampling the audio may help reduce processing time.
3. **Normalization**: Incorporating normalization for distance scores may help adjust for differences in audio length and volume.
4. **Plot Saving**: The plot\_mfcc() function could be updated to save images directly to a folder instead of just displaying them.
5. **Preprocessing**: Trimming silence and using noise reduction would likely improve DTW accuracy.

**Skills Gained**

Through this milestone, I improved my understanding of time-series comparison algorithms, audio signal processing, and the challenges of real-time audio analysis. I also learned how to use Python libraries like librosa, matplotlib, and scipy in combination to analyze and visualize data.

**Conclusion**

While the results were not as accurate as anticipated, this milestone successfully demonstrates how DTW can be used for audio comparison. The discrepancies between expected and actual results provide valuable feedback that can guide future improvements. The foundational structure is now in place, and with the right optimizations, this tool could evolve into a powerful application for audio similarity analysis.

**Figures**

A screenshot of a computer program

AI-generated content may be incorrect.

A graph with a number of orange squares

AI-generated content may be incorrect.

A graph with orange squares

AI-generated content may be incorrect.

A graph with a red line

AI-generated content may be incorrect.

**References**

Ellis, D. (n.d.). *Dynamic Time Warping (DTW)*. LabROSA - Columbia University. <http://labrosa.ee.columbia.edu/matlab/dtw/>

Librosa. (n.d.). *Librosa documentation*. <https://librosa.org/doc/main/index.html>

Salvador, S., & Chan, P. (2007). FastDTW: Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis, 11*(5), 561–580. <https://doi.org/10.3233/IDA-2007-11508>

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van Mulbregt, P. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods, 17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2