

Music Informatics

Final Projects – Music Recommendation System

Points: 3.8 out of 5

(Assignments and Final Projects contribute 50% to the final grade)

Due date: July 9th – 15:15 (before the exam)

**** In-class presentation is mandatory, otherwise the project will not be graded ****

Presentations will take place during the second half of the exam session.

Broad description of the Project & Tasks

Each team has 3 members. Each member will be responsible for one of the feature sets related to 3A, 3B, and 3C. All team members will equally contribute to the overall recommendation system.

1. Data Acquisition

Each member selects 10 of their top-10 tracks to analyze

In total you will be using 30 tracks / system (team)

If, due to space limitations, you cannot upload your tracks on eClass, you may send them to me through WeTransfer, for example.

2. Feature Extraction

- Implement your own algorithms or use “built-in” functions from libraries to extract the relevant audio features from your audio files (see also task 3)
- Do some statistical analyses on your frame-based features (e.g. mean, median, standard deviation)
- Provide visualizations of the statistical analyses with respect to the most important extracted features

3. Must explore features related to:

A. Harmony and/or Melody

- Chromagramms & Key detection (e.g. Minor vs. Major scale – **see the provided templates**)
- Chromagrams after voice/source separation (e.g., use Demucs, or Spotify’s algorithm)
- Pitch detection (e.g., using autocorrelation, YIN, etc.)

B. Rhythm & Tempo

- Tempo extraction (use a specific algorithm & provide parameter settings)
- May use metadata related to meter (e.g., 3/4, 4/4)

C. Timbre & Spectral Shape

- MFCCs (experiment with 13 to 20 coeffs)
- Timbre descriptors (related e.g., to overall voice quality)

4. Employ similarity metrics such as:

- Euclidean distance
- Cosine similarity
- Manhattan distance (useful for key-detection?)

5. Music Recommendation System

Using the similarity metrics between features, write a function that recommends the most similar and dissimilar music pieces to a given query piece. Use the 3 nearest/farthest neighbors.

- Must be able to demonstrate the recommendation function on the fly (in-class).
- May use a rule-based system using thresholds of specific feature values to recommend tracks
- May choose the most 2 or 3 dominant features and create a 2D or 3D plot of the tracks with respect to distance metrics
- Provide any visualizations that are helpful to understand how the recommender system works

6. Deliverables (Code & Report... don't forget the tracks!)

A. Code

(If I can't run your code the project will not be graded – Include all dependencies... if any!)

B. Implementation details of the music comparison system

Report on the following:

- The features you used, how you extracted them, and their justification related to the task
- The stats you used for feature time-series analysis (e.g., median vs. mean)
- The similarity metrics you used for each feature
- The quality of your results after experimenting for example with: different features, the thresholds related to the algorithms/functions you used for feature extraction, and the choice of stats
- The effectiveness of your approach (based on your own observations with respect to the dataset... and the expected constructive criticism from class)