MIREX SUBMISSION: SEQUENTIAL COMPLEXITY AS A DESCRIPTOR FOR MUSICAL SIMILARITY

Peter Foster, György Fazekas, Matthias Mauch, Simon Dixon
Centre for Digital Music
Queen Mary University of London
United Kingdom
{peter.foster,gyorgy.fazekas,matthias.mauch,simon.dixon}@eecs.qmul.ac.uk

ABSTRACT

In this submission, audio descriptors which quantify sequential complexity are used to predict musical similarity between pairs of tracks. We consider a data-driven approach for combining distances, where we estimate a regularised linear regression model.

1. INTRODUCTION

Our work in [2] forms the basis of this submission. The system models audio as track-wise summary statistics computed on frame-based features; Across considered audio features we then compute pairwise distance measures between statistics. To predict musical similarity, we combine pairwise distances using a linear model and then apply distance normalisation.

2. FEATURE EXTRACTION

For each track excerpt in the dataset, we extract a set of 25 audio features, using MIRToolbox [6] version 1.3.2 and using the framewise chromagram representation proposed by Ellis and Poliner [1]. With the exception of rhythmic features, which are computed using predicted onsets, features are based on a constant frame rate of 40Hz. Table 1 summarises the set of evaluated audio features.

3. FEATURE DESCRIPTORS

As a means of quantifying the sequential complexity of the audio feature vector sequence $\mathbf{V} = (\vec{v}_1, \dots, \vec{v}_T)$, we compute the compression rate $R_\lambda(\mathbf{V})$,

$$R_\lambda(\mathbf{V}) = \frac{C(\mathbf{V}, \lambda)}{T} \tag{1}$$

where $C(\mathbf{V}, \lambda)$ denotes the number of bits required to represent $\mathbf{V}$, given a quantisation scheme with λ levels and using a specified sequential compression scheme.

Feature name	Description
Chroma	12-component chromagram based on using phase-derivatives to identify tonal components in spectrum [1].
dynamics.rms	Root mean square of amplitude.
rhythm.tempo	Tempo estimate based on selecting peaks from autocorrelated onsets.
rhythm.attack.time	Duration of onset attack phase.
rhythm.attack.slope	Slope of onset attack phase.
spectral.centroid	First moment of magnitude spectrum.
spectral.brightness	Proportion of spectral energy above 1500Hz.
spectral.spread	Second moment of magnitude spectrum.
spectral.skewness	Skewness coefficient of magnitude spectrum.
spectral.kurtosis	Excess kurtosis of magnitude spectrum.
spectral.rolloff95	95th percentile of energy contained in magnitude spectrum.
spectral.rolloff85	85th percentile of energy contained in magnitude spectrum.
spectral.spectentropy	Shannon entropy of magnitude spectrum.
spectral.flatness	Wiener entropy of magnitude spectrum.
spectral.roughness	Average roughness [8] between peak pairs in magnitude spectrum.
spectral.irregularity	Squared amplitude difference between successive partials [5].
spectral.mfcc	12-component MFCCs [12] (excluding energy coefficient).
spectral.dmfcc	First-order differentiated MFCCs.
spectral.ddmfcc	Second-order differentiated MFCCs.
timbre.zerocross	Zero crossing rate.
timbre.spectralflux	Half-wave rectified L1 distance between magnitude spectrum at successive frames [7].
tonal.chromagram.centroid	Centroid of 12-component chromagram.
tonal.keyclarity	Peak correlation of chromagram with key profiles [3].
tonal.mode	Predicted mode after correlating chromagram with key profiles.
tonal.hcdf	Flux of 6-dimensional tonal centroid [4].

Table 1. Summary of evaluated audio features.

For each track, we compute compression rates on feature sequences extracted from musical audio. We refer to the set of compression rates as *feature complexity descriptors* (FCDs). For features based on constant frame rate, we compute FCDs using the original feature sequence, in addition to FCDs computed on downsampled versions of the original sequence. We consider the downsampling factors $\{1, 2, 4, 8\}$. In addition to FCDs, for each track excerpt we compute the mean and standard deviation, based on frame-level representation with no downsampling applied. We refer to such a 'bag-of-features' representation as *feature moment descriptors* (FMDs).

4. DISTANCE MEASURES

We compute Euclidean distances and symmetrised Kullback-Leibler (KL) divergences using 25 audio features and across both descriptor classes: For each pair of tracks, we obtain a total of 4×25 distances by computing Euclidean distances between FCDs at 4 temporal resolutions; we obtain a total of 2×25 distances by computing Euclidean distances and KL divergences between FMDs.

5. PREDICTING SIMILARITY

We predict musical similarity by computing a linear combination of distances. We obtain our linear model by applying regularised regression to annotated pairwise similarities, as described in [2]. Our submission differs as follows: We obtain audio and web-sourced tag annotations for approximately 10 000 tracks, sampled to maintain diversity of genres and artists. We then apply latent semantic analysis (LSA) to tag annotations, based on the method described in [10]. We consider a projection of tag annotations onto pairwise similarities between tracks as our response variable, which we seek to model. Using the response variable, we apply L2-regularised linear regression to pairwise distances between descriptors, based on the Matlab GLMNET library [1] .

6. NORMALISATION

To compensate for tracks consistently deemed similar to queries, we apply a two-step process. In the first step, we normalise predicted pairwise similarities by computing z-scores, as described in [9]. In the second step, we compute mutual proximity with independent Gaussian distributions, using the implementation described in [11].

7. REFERENCES

[1] D. P. W. Ellis and G.E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. IEEE Intern. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, pages 1429–1432, 2007.

[2] Peter Foster, Matthias Mauch, and Simon Dixon. Sequential complexity as a descriptor for musical similarity. *arXiv preprint arXiv:1402.6926*, 2014.

[3] E. Gómez. *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[4] Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proc. 1st ACM workshop on Audio and music computing multimedia*, pages 21–26. ACM, 2006.

[5] K. Jensen. *Timbre models of musical sounds*. PhD thesis, University of Copenhagen, Denmark, 1999.

[6] O. Lartillot and P. Toiviainen. A Matlab toolbox for musical feature extraction from audio. In *Proc. Intern. Conf. Digital Audio Effects (DAFx)*, pages 237–244, 2007.

[7] P. Masri. *Computer modelling of sound for transformation and synthesis of musical signals*. PhD thesis, University of Bristol, United Kingdom, 1996.

[8] R. Plomp and W.J.M. Levelt. Tonal consonance and critical bandwidth. *Journal of the Acoustical Society of America*, 38:548, 1965.

[9] Suman Ravuri and Daniel PW Ellis. Cover song detection: from high scores to general classification. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 65–68. IEEE, 2010.

[10] Pasi Saari, Mathieu Barthet, Gyorgy Fazekas, Tuomas Eerola, and Mark Sandler. Semantic models of musical mood: Comparison between crowd-sourced and curated editorial tags. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[11] Dominik Schnitzer, Arthur Flexer, Markus Schedl, and Gerhard Widmer. Local and global scaling reduce hubs in space. *The Journal of Machine Learning Research*, 13(1):2871–2902, 2012.

[12] M. Slaney. Auditory toolbox version 2. Technical report, Interval Research Corporation, 1998.

[1] `http://www.stanford.edu/~hastie/glmnet_matlab/`