LEARNING SONG SIMILARITY FROM RADIO STATION PLAYLISTS

François Maillet
Department of Computer Science
Université de Montréal
Montreal, Canada
mailletf@iro.umontreal.ca

Douglas Eck
Department of Computer Science
Université de Montréal
Montreal, Canada
eckdoug@iro.umontreal.ca

ABSTRACT

This extended abstract details a submission to the 2009 Music Information Retrieval eXchange (MIREX) audio music similarity and retrieval task.

We propose a method for computing song similarity by learning song transition probabilities from audio features extracted from songs played in professional radio station playlists. We train a binary classifier to distinguish between 2-song sequences that were played on the radio and sequences that were not. The certainty of the trained model when presented with a song sequence can then be interpreted as the similarity between the two songs.

The model used is explained in more details in [1].

1. LEARNING FROM RADIO STATION PLAYLISTS

1.1 Defining a playlist

Our similarity model is trained on professional radio station playlists. For this experiment, we consider a playlist to be a sequence of 2 consecutive plays uninterrupted by a commercial break. Suppose a radio station plays the tracks t_a, t_b and t_c one after the other, we will consider $\{t_a, t_b\}$ and $\{t_b, t_c\}$ as two 2-song sequences $\in \mathbb{S}^2$. We consider the sequences $\{t_a, t_b\}$ and $\{t_b, t_a\}$ as two distinct sequences. The model's output will thus be non-symmetric in regards to the order in which the songs are presented.

1.2 Playlist sources

We used playlist data from two sources.

First, we used data from the free Internet-streamed radio station RadioParadise [1], who provided us with 575 days worth of data. The data consists of 195,692 plays, 6,328 unique songs and 1,972 unique artists.

We also used Yes.com's API [2], which gives access to the play history of thousands of radio stations in the United States. We mined data for 57 days, totaling in 449 stations, 6,706,830 plays, 42,027 unique songs and 9,990 unique artists.

[1] http://www.radioparadise.com
[2] http://api.yes.com

1.3 Putting the data together

Combining all the data yielded 6,902,522 plays, with an average of 15,338 plays per station. Since our model uses audio features as input, we need the audio file for all the songs we will use. Of the 47,044 total songs played in the playlists we used, we were able to obtain the audio files for 7,127 tracks. This reduced the number of distinct usable 2-song sequences to 180,232. The sequences for which we had all the audio files were combinations from 5,562 tracks.

We did not possess a set of explicit negative examples (i.e. two-song sequences that a radio station would never play). In order to train the model with examples from both the positive and negative class, we considered any song sequence that was not observed as a negative example. During training, at each new epoch, we randomly sampled a new set of negative examples matched in size to our positive example set.

2. SIMILARITY MODEL OVERVIEW

2.1 Features

We use audio-based features as input to our model. First, we compute 176 frame-level autocorrelation coefficients for lags spanning from 250ms to 2000ms at 10ms intervals. These are aggregated by simply taking their mean. We then down sample the values by a factor of two, yielding 88 values. We then take the first 12 Mel-frequency cepstral coefficients (MFCC), calculated over short windows of audio (100ms with 25ms overlaps), and model them with a single Gaussian (G1) with full covariance [3]. We unwrap the values into a vector, which yields 78 values.

We then compute two song-level features, danceability [5] and long-term loudness level (LLML) [4]. Danceability is a variation of detrended fluctuation analysis, which indicates if a strong and steady beat is present in the track, while the LLML gives an indication of the perceived loudness of the track. Both of these features yield a single numeric value per song.

These 4 audio features are concatenated to form an 180 dimensional vector for each track.

2.2 Learning models

We compare two types of learning models for this task: multi-layer perceptrons (MLP) and stacked denoising auto-encoders (SdA).

The first model, the MLP, a 3-layer feedforward neural network, is a non-linearity classifier that learns a compact, nonlinear set of basis functions.

We also use a type of deep neural network called a stacked denoising auto-encoder (SdA) [2]. The SdA learns a hierarchical representation of the input data by successively initializing each of its layers according to an unsupervised criterion to form more complex and abstract features. The goal of this per-layer unsupervised learning is to extract an intermediate representation which preserves information content whilst being invariant to certain transformations in the input. SdAs are exactly like neural networks with the exception that they have multiple hidden layers that are initialized with unsupervised training.

In our experiments, the models operated directly on pairs of audio features. The input of our model is thus a vector of length $180 \cdot 2$, formed by concatenating the features of each track into a single vector.

3. MIREX TASK

The audio music similarity and retrieval task involves computing a full track-to-track similarity matrix for 7,000 30-second audio clips. The clips are in 22.05kHz mono wav format and come from 10 musical genres.

We use the radio station playlist dataset we collected as detailed in Section 1 to train an MLP and an SdA-3 model. When presented with the concatenated features of two songs, the models' output represents the probability of those two songs following one another in a playlist. The output can also be interpreted as the similarity between the two songs.

We compute the audio features as described in Section 2.1 for the 7,000 tracks and use our trained models to compute the track-to-track similarity matrix.

4. REFERENCES

[1] F. Maillet, D. Eck, G. Desjardins, P. Lamere: "Steerable Playlist Generation by Learning Song Similarity from Radio Station Playlists," *Proceedings of the 10th International Conference on Music Information Retrieval*, Kobe, Japan, 2009

[2] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol: "Extracting and Composing Robust Features with Denoising Autoencoders," *International Conference on Machine Learning*, 2008

[3] M. Mandel, D. Ellis: "Song-Level Features and Support Vector Machines for Music Classication," *Proceedings of the 6th International Conference on Music Information Retrieval*, London, UK, 2005.

[4] E. Vickers: "Automatic long-term loudness and dynamics matching," *Proceedings of the AES 111th Convention*, New York, USA, 2001.

[5] S. Streich: "Music Complexity: a multi-faceted description of audio content," Ph.D. Dissertation, UPF, Barcelona, 2007.