DP1, MP1, MP2 ENTRIES FOR MIREX2013 STRUCTURAL SEGMENTATION AND BEAT TRACKING

Brian McFee
Center for Jazz Studies
Columbia University
brm2132@columbia.edu

Daniel P.W. Ellis
Electrical Engineering
Columbia University
dpwe@columbia.edu

Table 1. Algorithm parameters and settings.

Parameter	Symbol	Value
Sampling rate (Hz)	*sr*	22050
FFT size	N	2048
Hop size	h	64
# Mel bins	d_M	128
Max frequency (Hz)	$f_{\max}$	8000
# MFCCs	d_t	32
# Chroma	d_p	12
# Repetition features	d_r	32
Repetition window (beats)	δ	3
Repetition filter (beats)	w	7
Repetition links (beats)	k	$2\sqrt{n}$
Minimum number of segments	$K_{\min}$	8
Maximum number of segments	$K_{\max}$	24

ABSTRACT

This extended abstract describes the algorithms for beat tracking and structural segmentation submitted as DP1 and MP1–2 to MIREX2013.

1. INTRODUCTION

This document describes the implementation details for MIREX2013 submissions DP1 (beat tracking) and MP1–2 (structural segmentation). All algorithms are implemented in Python, and code will be made publicly available at the end of the evaluation.

Section 2 describes the beat tracking method (DP1), which is subsequently used in the segmentation method described in Section 3 (MP1). Section 4 (DP2) describes a refinement of the MP1 method which uses a supervised learning technique to optimally combine features for segmentation.

Table 1 includes a summary of the parameters of the various components.

2. DP1: BEAT TRACKING

The DP1 beat tracker is based on the dynamic programming method described by Ellis [3]. The submitted version is based on the Python implementation provided as part of the `librosa`[1] package, which differs from the original matlab implementation by including a frame offset correction to compensate for the systemic lag observed by Davies *et al.* [2].

The beat tracker operates by solving a dynamic program to select peaks from an onset strength function. Given a log-magnitude Mel spectrogram M, [3] defines an onset strength function as the sum over frequency bands of thresholded spectral difference:

$$\omega_0(t) = \sum_f \max(0, M_{f,t} - M_{f,t-1}).$$

The current implementation differs from the original in two ways. First, the frequency resolution has been increasing the sampling rate *sr* from 8000 to 22050, FFT size N from 256 to 2048, and the number of Mel bins d_M from 40 to 128 (see Table 1). Second, the onset detection function was modified to prefer vertical synchronicity of onsets across frequency bins, rather than raw (log-) magnitude, by replacing the sum operator with a median:

$$\omega(t) = \operatorname*{median}_f \max(0, M_{f,t} - M_{f,t-1}).$$

The resulting onset strength function tends to be sparser, and less sensitive to high-magnitude asynchronous events. In practice, we have observed significant improvements in accuracy over the previous method.

3. MP1: UNSUPERVISED STRUCTURAL SEGMENTATION

The submitted methods for structural segmentation are based upon agglomerative clustering of beat-synchronous feature descriptors, similar to the method of Levy and Sandler [6].

For a given song, the general strategy is as follows:

1. Detect beats using the method of Section 2;
2. compute a beat-synchronous feature matrix $X \in \mathbb{R}^{D \times n}$ where D denotes the number of features, and n denotes the number of beats;

[1] `http://github.com/bmcfee/librosa`

3. perform temporally-constrained agglomerative clustering on the columns of X;

4. prune the cluster tree to K segments.

3.1 Features

For each song, we compute a variety of both local and global beat-level features, as follows.

3.1.1 Timbre and pitch

As a first step, we compute the top $d_M = 32$ Mel frequency cepstral coefficients (MFCC) and chroma vector for each audio frame. Across frames within each beat, the MFCCs are mean-aggregated, and the chroma vectors are median-aggregated.

3.1.2 Repetition

In addition the local timbre and pitch descriptors described above, we encode global repetition structure. The method is inspired by the method of Serrà *et al.* [8], with several modifications.

The proposed method first computes a (possibly asymmetric) k-nearest-neighbor graph over beat-synchronous timbre descriptors X, resulting in a binary recurrence matrix R, where

$$R_{i,j} = 1 \Leftrightarrow X_j \text{ is a nearest neighbor of } X_i.$$

Links within $\pm\delta = \pm 3$ beats of i are suppressed, as they do not contribute meaningful global structure information. Each beat is connected to $k = 2\sqrt{n}$ nearest neighbors.

The recurrence matrix is then skewed so that

$$S_{i,j} = 1 \Leftrightarrow R_{i,j+i} = 1,$$

effectively reparameterizing the recurrence matrix from time-time to time-lag. Note that unlike [8], we do not wrap positive- and negative-lag on top of each-other, resulting in a taller matrix $S \in \{0,1\}^{2n,n}$.

To correct for linkage errors, S must be blurred or smoothed in some way. Rather than applying a 2-dimensional Gaussian filter, we first observe that the use of beat-synchronous features significantly reduces the effects of tempo variation, so a vertical smoothing is unnecessary. Instead, we apply a 1-dimensional median filter of width $w = 7$ to each row of S. The median filter more cleanly preserves boundaries, and eliminates binary noise (skipped beats and spurious linkages). The width of the filter also carries an intuitive interpretation as the minimum segment duration to detect.

As a final step, the filtered time-lag matrix S' is compressed to a latent structure representation by singular value decomposition. Let $S' = U\Sigma V^\mathsf{T}$ denote the SVD; since U is unitary, applying the transformation $S' \mapsto U^\mathsf{T} S'$ will not change distances or the induced clustering of columns. Similarly, normalizing by the maximal singular value σ_1 does not change the induced clustering, but ensures that the latent repetition features lie within a bounded region, regardless of the number of beats. We therefore apply the following transformation:

$$S' \to \sigma_1^{-1} U^\mathsf{T} S',$$

and project onto the top-$d_r = 32$ dimensions. While the low-rank projection is not strictly necessary when segmenting a single song, it ensures that the representation lies in a fixed-dimensional space, and is critical when learning a feature weighting from multiple songs (Section 4).

Using the method described above, we compute two sets of latent structure features for each song: one based on timbre similarity using the beat-synchronous MFCCs, and one based on history-stacked beat-synchronous chroma vectors.

3.1.3 Time and location

Finally, we append the following four features to each beat: i) time (in seconds) of the beat, ii) normalized time of the beat, iii) raw index of the beat (*e.g.*, 0, 1, 2, ...), iv) normalized beat index. These features add an implicit regularization to the clustering algorithm by adding a quadratic penalty to the duration of each detected segment.

The final feature matrix X consists of $D = d_t + d_p + 2d_r + 4 = 112$ features for each beat.

3.2 Clustering and pruning

Given a feature matrix $X \in \mathbb{R}^{D \times n}$, the columns are recursively merged via linkage-constrained agglomerative clustering. Specifically, we apply the Scikit-learn implementation of Ward's method [7, 10], with a constraints of the form $(t, t+1)$ for all $0 \leq t < n$.

The output of the clustering algorithm on X is a cluster tree, *i.e.*, a dendrogram over cluster refinements induced by the iterative merging procedure. The tree therefore encodes clusterings for all numbers of clusters $1 \leq K \leq n$. The cluster tree is first simplified by pruning to at least $K_{\min} = 8$ and at most $K_{\max} = 24$ clusters.

The final pruning of the resulting cluster tree is selected by optimizing an AIC-corrected score function [1]. Specifically, for a potential number of clusters K, the corresponding clustering is selected from the tree, and for each segment $1 \leq i \leq K$, the centroid μ_i is estimated. This can be interpreted as fitting a sequence of isotropic Gaussian distributions $\mathcal{N}(\mu_i, I)$ to the selected columns $X_{[i]}$. Given this interpretation, we compute the average log-likelihood of the data ($X_{[i]}$) under each segment's model, and add the AIC correction $-K \cdot D$. The optimal value is selected by searching over $K_{\min} \leq K \leq K_{\max}$.

Given the final clustering of beat-synchronous features, the segment boundaries are detected by locating cluster-label disagreements, and converting back to the time index of the corresponding beat frames. The 0-marker and end-of-track time are included in the final segment predictions.

4. MP2: SUPERVISED SEGMENTATION

The quality of the clustering produced in the previous section will ultimately depend on the computation of distances

between columns of X. In this section, we propose a method to optimize the distance function to improve clustering.

The proposed method is based upon the multi-class variant of Fisher's linear discriminant analysis (FDA) [4,5]. In the context of segmentation, we consider a "class" to be a segment identifier (though not it's structural label, so two different "chorus" segments map to two different classes). A structural segmentation annotation for a song can be converted into a labeled data set by first extracting the feature matrix $X \in \mathbb{R}^{D \times n}$, and then assigning each column X_i to the segment y_i which contains it. In the usual multi-class FDA setting, the data would be transformed via the learned mapping $X \mapsto W^\mathsf{T} X$, where

$$W := \underset{W}{\operatorname{argmin}} \operatorname{tr}\left((W^\mathsf{T} S_\mathrm{B} W)^{-1} W^\mathsf{T} S_\mathrm{W} W\right), \quad (1)$$

where S_W and S_B are the *within-* and *between*-class scatter matrices:

$$S_\mathrm{W} := \sum_k \sum_{i:y_i=k} (X_i - \mu_k)(X_i - \mu_k)^\mathsf{T}$$
$$S_\mathrm{W} := \sum_k n_k (\mu_k - \mu)(\mu_k - \mu)^\mathsf{T},$$

and n_k denotes the number of beats in segment k. Equation (1) is equivalent to a generalized eigenvalue problem, and can thus be solved efficiently. Moreover, the scatter matrices from multiple songs can be accumulated into S_W and S_B, allowing the method to generalize across large collections.

FDA generally tries to condense points within each class (segment), and spread the centroids of each class (segment) far apart from each-other. This would have the undesirable effect of attempting to spread the centroids of, *e.g.*, two chorus segments far apart from each-other. To counteract this effect, we modified the FDA objective to only spread means between sequentially adjacent segments $(k, k+1)$. The resulting *ordinal* FDA algorithm optimizes

$$W := \underset{W}{\operatorname{argmin}} \operatorname{tr}\left((W^\mathsf{T}(S_\mathrm{O} + \lambda I) W)^{-1} W^\mathsf{T} S_\mathrm{W} W\right), \quad (2)$$

where

$$S_\mathrm{O} := \sum_{k<K} n_k (\mu_k - \mu_{k+})(\mu_k - \mu_{k+})^\mathsf{T}$$
$$\mu_{k+} := \frac{n_k \mu_k + n_{k+1}\mu_{k+1}}{n_k + n_{k+1}},$$

and $\lambda > 0$ is a regularization term which is selected by cross-validation on a labeled training set. As in FDA, scatter matrices are aggregated across multiple songs.

In order to set λ and learn W, we used a portion of the SALAMI dataset which is publicly available on `http://archive.org/` [9]. All annotations were extracted from the `parsed/textfile1_functions.txt` files.

After applying the learned transformation $X \mapsto W^\mathsf{T} X$, the clustering and pruning steps follow just as in MP1.

5. REFERENCES

[1] Hirotugu Akaike. Information theory and an extension of the maximum likelihood principle. In *Second international symposium on information theory*, pages 267–281. Akademinai Kiado, 1973.

[2] Matthew EP Davies, Norberto Degara, and Mark D Plumbley. Evaluation methods for musical audio beat tracking algorithms. 2009.

[3] Daniel PW Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[4] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[5] Keinosuke Fukunaga. *Introduction to statistical pattern recognition*. Access Online via Elsevier, 1990.

[6] Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):318–326, 2008.

[7] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.

[8] Joan Serra, Meinard Müller, Peter Grosche, and Josep Lluis Arcos. Unsupervised detection of music boundaries by time series structure features. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[9] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie. Design and creation of a large-scale database of structural annotations. In *ISMIR*, pages 555–560, 2011.

[10] Joe H Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.