LEARNING TO SEGMENT SONGS WITH ORDINAL LINEAR DISCRIMINANT ANALYSIS

Brian McFee

Center for Jazz Studies
Columbia University
brm2132@columbia.edu

Daniel P.W. Ellis

LabROSA, Department of Electrical Engineering
Columbia University
dpwe@columbia.edu

ABSTRACT

This paper describes a supervised learning algorithm which optimizes a feature representation for temporally constrained clustering. The proposed method is applied to music segmentation, in which a song is partitioned into functional or locally homogeneous segments (*e.g.*, verse or chorus). To facilitate abstraction over multiple training examples, we develop a latent structural repetition feature, which summarizes the repetitive structure of a song of any length in a fixed-dimensional representation. Experimental results demonstrate that the proposed method efficiently integrates heterogeneous features, and improves segmentation accuracy.

Index Terms— Music, automatic segmentation, learning

1. INTRODUCTION

Automatic music segmentation algorithms take as input the acoustic signal of a musical performance, and produce a temporal partitioning of the performance into a small number of *segments*. Ideally, segments correspond to structurally meaningful regions of the performance, such *verse* or *chorus*.

Common approaches to music segmentation attempt to detect repeated patterns of features (*e.g.*, a repeating chord progression), often by some form of clustering [1] or novelty detection [2]. Often, features are manually tuned and optimized for a specific development set, and carry implicit assumptions about the nature of musical structure. As a concrete example, features built to detect repeated chord progressions may work well for characterizing some genres (*e.g.*, rock or pop), but fail for other styles (*e.g.*, jazz or hip-hop) which may be structured around timbre rather than melody.

In this work, we propose a supervised learning algorithm to automatically adapt acoustic and structural features to the statistics of a training set. Given a collection of songs with structural annotations, the algorithm finds an optimal linear transformation of features to preserve and predict segments.

This work was supported by a grant from the Mellon foundation, and grant IIS-1117015 from the National Science Foundation (NSF).

1.1. Our contributions

Our primary contribution in this work is the ordinal linear discriminant analysis (OLDA) technique to learn a feature transformation which is optimized for musical segmentation, or more generally, time-series clustering. As a secondary contribution, we propose a latent structural repetition descriptor, which facilitates learning and generalization across multiple examples.

1.2. Related work

The segmentation algorithm we use is most similar to the constrained clustering method of Levy and Sandler [1], which incorporated sequential consistency constraints to a hidden Markov model. The method proposed here is simpler, and uses a sequentially constrained agglomerative clustering algorithm to produce a hierarchical segmentation over the entire track. Because the segmentation is hierarchical, the number of segments need not be specified in advance.

The proposed latent repetition features are adapted from the work of Serrà *et al.* [2]. While qualitatively similar, we apply different filtering and beat synchronization techniques to better preserve segment boundaries. In addition to chord sequence repetitions, our method includes timbre repetitions, as well as localized timbre, pitch, and timing information.

2. MUSIC SEGMENTATION

The criteria for deciding what is or is not a segment may vary across genres or styles. Pop music relies heavily on a verse/chorus structure, and is well characterized by repeating chord sequences. On the other hand, jazz tends to be structured by changing instrumentation (*e.g.*, the current soloist), and is better modeled as sequences of consistent timbre. In general, a structural segmentation algorithm should include multiple feature representations in order to function on various musical genres.

As a first step toward integrating multiple features, we introduce a structural repetition feature which is amenable to learning and abstraction across multiple example songs.

2.1. Latent structural repetition

Figure 1 outlines our approach for computing structural repetition features, which is adapted from Serrà *et al.* [2]. First, we extract beat-synchronous features (*e.g.*, MFCCs or chroma) from the signal, and build a binary self-similarity matrix by linking each beat to its nearest neighbors in feature space (fig. 1, top-left). With beat-synchronous features, repeated sections appear as diagonals in the self-similarity matrix. To detect repeated sections, the matrix is skewed by shifting the ith column down by i rows (fig. 1, top-right), thereby converting diagonals into horizontals.

Nearest-neighbor linkage can result spurious links and skipped connections. Serrà *et al.* resolve this by convolving with a Gaussian filter, which suppresses noise, but also blurs segment boundaries. Instead, we use a horizontal median filter, which (for odd window length) produces a binary matrix, suppresses links outside of repeated sequences, and fills in skipped connections (fig. 1, bottom-left). The width of the median filter directly corresponds to the minimal duration (in beats) of detected repetitions, and is consequently easier to tune than a Gaussian filter. The median filter also preserves edges better than the Gaussian filter, so we may expect more precise detection of segment boundaries.

Let $R \in \mathbb{R}^{2t \times t}$ denote the median-filtered, skewed self-similarity matrix over t beats. Because the dimensionality (number of rows) of R varies from one track to the next, it is difficult to model and generalize across collections. However, what matters for segmentation is not the representation of the columns of R, but the similarity between them. More precisely, methods which depend on distances between column-vectors — *i.e.*, those based on clustering or novelty curves — are invariant to unitary transformations U^T: $\|R_{\cdot,i} - R_{\cdot,j}\| = \|U^\mathsf{T} R_{\cdot,i} - U^\mathsf{T} R_{\cdot,j}\|$.

We therefore introduce *latent structural repetition*, which compresses each any song's R matrix to a fixed-dimension representation. Let $R = U \Sigma V^\mathsf{T}$ denote the singular value decomposition of R, with (descending) singular values σ_i. The latent structural repetition feature is defined as the matrix L:

$$L := \sigma_1^{-1} U^\mathsf{T} R = \sigma_1^{-1} \Sigma V^\mathsf{T}. \tag{1}$$

Reducing L to $d < 2t$ principal components retains the most important factors, and normalizing by σ_1 reduces the influence of track duration. Figure 1 (bottom-right) depicts an example of the resulting features. Small values of d often suffice to capture global structure: in the given example, the top component suffices to detect transitions between verse (non-repetitive) and chorus (repetitive).

2.2. Constrained agglomerative clustering

Given a feature matrix $X \in \mathbb{R}^{D \times t}$, we produce a hierarchical clustering of the columns of X by using the linkage-constrained variant of Ward's agglomerative clustering algorithm [3] as implemented in `scikit-learn` [4]. For each column $X_{\cdot,i}$, linkage constraints are generated for $(i-1, i)$ and $(i, i+1)$. Starting from t clusters (one for each column of X), linked clusters are iteratively merged until only one remains. The hierarchy is computed in $\mathcal{O}(t)$ merge operations, and due to the constraints, there are $\mathcal{O}(t)$ feasible merges at each step. Each merge operation takes time $\mathcal{O}(D + \log t)$ (to compute centroids and manage a priority queue), so the algorithm runs in $\mathcal{O}(tD + t \log t)$ time. For $D \in \Omega(\log t)$, the cost of clustering is dominated by the $\Omega(tD)$ cost of computing X.

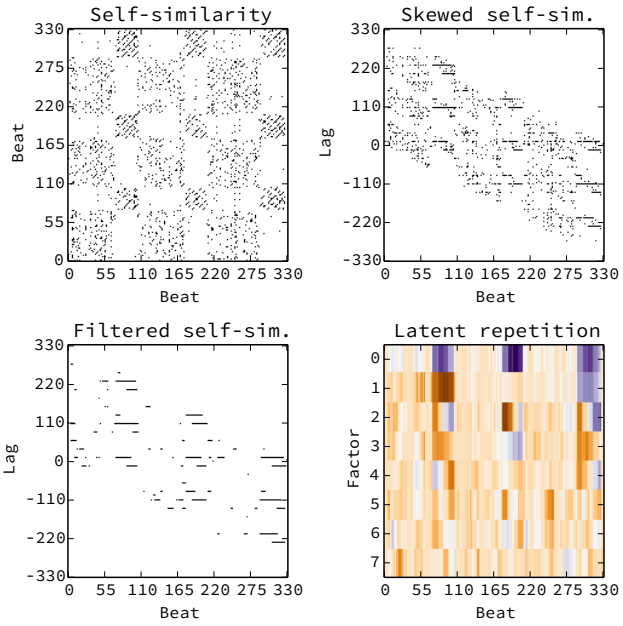

Fig. 1. Repetition features derived from *Tupac Shakur — Trapped*. Top-left: a binary self-similarity (k-nearest-neighbor) matrix over beats. Top-right: the time-lag transformation of the self-similarity matrix. Bottom-left: the result of the horizontal median filter. Bottom-right: 8-dimensional latent factor representation (best viewed in color).

2.3. Choosing the number of segments

The hierarchical clustering produces segmentations for all numbers of segments $1 \leq k \leq t$. Because music segmentation is itself an ambiguous task, the ability to simultaneously produce segmentations at all resolutions is a key advantage of the proposed technique. However, standard evaluation procedures are defined only for flat segmentations, and require a specific pruning of the segment hierarchy.

To select the number of segments k, we compute the clustering cost of each pruning within a plausible bounded range $k_{\min} \leq k \leq k_{\max}$. The bounds are determined by assuming average minimum and maximum segment duration of 10s and 45s. AIC correction [5] is then applied to each candidate pruning (assuming a spherical Gaussian model for each segment), and k is chosen to minimize the AIC-corrected cost.

2.4. Multiple features

Structural repetition features are most commonly used to detect repeated chord sequences, which is appropriate for segmenting many genres of popular music. However, the conventions of structure can vary from one genre to the next, so a general segmentation algorithm should include a variety of feature descriptors. We therefore aggregate several types of feature descriptors for each beat:

Timbre mean Mel-frequency cepstral coefficients,

Pitch (coordinate-wise) median chroma vectors,

Timbre repetition latent MFCC repetitions,

Pitch repetition latent chroma repetitions,

Time Time-stamp (in seconds) of each beat, normalized time-stamps (as a fraction of track duration), beat indices $(1, 2, \ldots, t)$, and normalized beat indices $(1/t, 2/t, \ldots, 1)$.

Local timbre features can be useful for non-repetitive forms (*e.g.*, jazz), while timbre repetition is useful for sample-based genres (*e.g.*, hip-hop or electonic). Similarly, chroma features capture local pitch similarity, while pitch repetitions capture chord progressions. The time features act as an implicit quadratic regularization on segment durations, and promote balanced segmentations. We include normalized and unnormalized time-stamps to allow the learning algorithm (Section 3) to adapt the regularization to either relative or absolute segment durations; beat index features differ from raw time features by correcting for tempo variation.

All features can be stacked together into a single feature matrix $X \in \mathbb{R}^{D \times t}$, and clustered via the method described above. However, the relative magnitude and importance of each feature is not calibrated to produce optimal clusterings.

3. ORDINAL LINEAR DISCRIMINANT ANALYSIS

To improve the feature representation for clustering, we propose a simple adaptation of Fisher's linear discriminant analysis (FDA) [6]. In its multi-class form, FDA takes as input a labeled collection of data $x_i \in \mathbb{R}^D$ and class labels $y_i \in \{1, 2, \ldots, C\}$, and produces a linear transformation $W \in \mathbb{R}^{D \times D}$ that simultaneously maximizes the distance between class centroids, and minimizes the variance of each class individually [7]. This is accomplished by solving the following optimization:

$$W \coloneqq \operatorname*{argmax}_W \operatorname{tr}\left(\left(W^\mathsf{T} A_\mathrm{W} W\right)^{-1} W^\mathsf{T} A_\mathrm{B} W \right), \qquad (2)$$

where A_W and A_B are the *within*- and *between*-class scatter matrices:

$$A_\mathrm{W} \coloneqq \sum_c \sum_{i:y_i=c} (x_i - \mu_c)(x_i - \mu_c)^\mathsf{T}$$

$$A_\mathrm{B} \coloneqq \sum_c n_c (\mu_c - \mu)(\mu_c - \mu)^\mathsf{T},$$

μ denotes the mean across all classes, μ_c is the mean of class c, and n_c denotes the number of examples in class c. Equation (2) can be efficiently solved as a generalized eigenvalue problem over the two scatter matrices $(A_\mathrm{B}, A_\mathrm{W})$ [8].

Class labels can be synthesized from segments on an annotated training song, so that the columns of X belonging to the first segment are assigned to class 1, the second segment to class 2, and so on. However, interpreting each segment as a distinct class could result in a repeated verse being treated as two distinct classes which cannot be separated. A more serious problem with this formulation is that it is unclear how to generalize across multiple songs, as it would result in FDA attempting to separate segments from different songs.

Due to linkage constraints, the agglomerative clustering algorithm (section 2.2) only considers merge operations over successive segments $(c, c+1)$. This motivates a relaxed FDA formulation which only attempts to separate adjacent segments. This is accomplished by replacing the between-class scatter matrix A_B with the resulting *ordinal scatter matrix*:

$$A_\mathrm{O} \coloneqq \sum_{c<C} n_c(\mu_c - \mu_{c+})(\mu_c - \mu_{c+})^\mathsf{T} + n_{c+1}(\mu_{c+1} - \mu_{c+})(\mu_{c+1} - \mu_{c+})^\mathsf{T}$$

$$\mu_{c+} \coloneqq \frac{n_c \mu_c + n_{c+1}\mu_{c+1}}{n_c + n_{c+1}}.$$

Intuitively, A_O measures the deviation of successive segments $(c, c+1)$ from their mutual centroid μ_{c+}, which is exactly the comparison performed for merge operations in the agglomerative clustering algorithm. Optimizing W to maximize this deviation, while minimizing within-segment variance, should enhance the overall segmentation accuracy.

To improve numerical stability when A_W is singular, we include a smoothing parameter $\lambda > 0$.[1] The OLDA optimization takes the form:

$$W \coloneqq \operatorname*{argmax}_W \operatorname{tr}\left(\left(W^\mathsf{T}(A_\mathrm{W} + \lambda I)W\right)^{-1} W^\mathsf{T} A_\mathrm{O} W \right), \quad (3)$$

which again can be solved efficiently as a generalized eigenvalue problem over the matrix pair $(A_\mathrm{O}, A_\mathrm{W} + \lambda I)$.

Because interactions are only measured between neighboring segments, it is straightforward to include data from multiple songs by summing their individual contributions to A_O and A_W. After learning W, the feature matrix X for a previously unseen song is transformed via $X \mapsto W^\mathsf{T} X$, and then clustered as described in Section 2.2.

4. EVALUATION

All proposed methods are implemented in Python with the `librosa` package.[2] All signals were downsampled to

[1] The same regularization strategy is applied to FDA in Section 4.

[2] Code is available at `https://github.com/bmcfee/olda`.

22KHz mono, and analyzed with a 93ms window and 3ms hop. MFCCs are generated from 128 Mel bands with an 8KHz cutoff. We take 32 MFCCs and 12 chroma bins; repetition features are calculated with $2\sqrt{t}$ nearest neighbors, median-filtered with a window width of 7, and projected to 32 dimensions each. Including the four time-stamp features, the combined representation has dimension $D = 112$. Beats were detected by the *median-percussive* method [9].

4.1. Data and metrics

To evaluate the proposed methods, we evaluate predicted segmentations on two publicly available datasets:

Beatles-ISO 179 songs by the Beatles [10, 11], and

SALAMI-free 253 songs from the SALAMI dataset [12] which are freely available on the Internet Archive [13].

Both datasets provide labels for each annotated segment (*e.g.*, *verse* or *intro*), but we ignore these labels in this set of experiments. Compared to the Beatles corpus, SALAMI consists of tracks by multiple artists, and has much more diversity of genre, style, and instrumentation.

On both datasets, we compare to SMGA [2], which achieved the highest performance in the 2012 MIREX structural segmentation evaluation [14]. On SALAMI-Free, we include comparisons to C-NMF [13] and SI-LPCA [15].

For both datasets, we evaluate the unweighted feature representation (*Unweighted*), FDA optimization (using the one-class-per-segment approach described in Section 3), and OLDA. To ensure fairness of evaluation, the FDA and OLDA models used on the Beatles-ISO were trained using only SALAMI-free data, and vice versa. FDA and OLDA were trained by optimizing $\lambda \in \{10^0, 10^1, \ldots, 10^9\}$ to maximize S_F score (see below) on the training set.

We report the following standard segmentation metrics: [3]

Boundary retrieval F-measure of segment boundary detection within a 0.5s or 3s window,

Normalized conditional entropy (NCE) Harmonic mean (S_F) of over- and under-segmentation S_O and S_U,

Frame clustering F-measure (F_C) of detecting whether any two frames belong to the same segment.

Note that boundary retrieval and frame clustering metrics are sensitive to the number of predicted segments, upon which multiple human annotators often disagree. NCE is designed to be robust to changes in segmentation granularity. Following MIREX practice, frames are sampled at a frequency of 10Hz for the NCE and frame clustering metrics.

Table 1. Segmentation performance. Best scores are indicated in bold; significance is assessed with a Bonferroni-corrected Wilcoxon signed-rank test at $\alpha = 0.05$.

Algorithm	$F_{0.5s}$	F_{3s}	S_F	F_C
Beatles-ISO				
Unweighted	0.264	0.497	0.789	0.662
FDA	**0.292**	0.537	0.807	0.685
OLDA	**0.306**	0.547	0.812	0.690
SMGA [2]	0.153	**0.658**	**0.829**	**0.729**
SALAMI-free				
Unweighted	0.200	0.427	0.791	0.622
FDA	**0.230**	0.467	**0.806**	**0.635**
OLDA	**0.240**	0.468	**0.807**	**0.636**
SMGA [2]	0.134	**0.508**	0.786	0.550
C-NMF [13]	0.110	0.463	0.767	0.550
SI-PLCA [15]	0.128	0.286	0.643	0.459

4.2. Results

Table 1 lists the results for the Beatles-ISO and SALAMI-free. SMGA performs best[4] on most metrics for Beatles-ISO. The NCE scores are qualitatively comparable between SMGA and OLDA (S_F of 0.829 and 0.812, respectively). The proposed methods achieve higher accuracy for boundary detection at 0.5s resolution, which can be attributed to the increased precision afforded by the beat-synchronous and median-filtered repetition features.

On both datasets, OLDA consistently improves over the unweighted model, and achieves the highest scores on three out of four metrics for SALAMI-free.

5. CONCLUSION

This paper introduced the ordinal linear discriminant analysis (OLDA) method for learning feature projections to improve time-series clustering. The proposed latent structural repetition features provide a convenient, fixed-dimensional representation of global song structure, which facilitates modeling across multiple songs. The effective combination of global structural cues with local features results in significant improvement in segmentation accuracy on the mixed-genre SALAMI-free dataset.

6. ACKNOWLEDGMENTS

The authors acknowledge support from The Andrew W. Mellon Foundation, and NSF grant IIS-1117015.

[3] For a detailed description of segmentation metrics, see [16].

[4] SMGA parameters were tuned to perform well on the Beatles data [2].

7. REFERENCES

[1] Mark Levy and Mark Sandler, "Structural segmentation of musical audio by constrained clustering," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 2, pp. 318–326, 2008.

[2] Joan Serrà, 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.

[3] Joe H Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.

[4] 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*, vol. 12, pp. 2825–2830, 2011.

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

[6] Ronald A Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

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

[8] Tijl De Bie, Nello Cristianini, and Roman Rosipal, "Eigenproblems in pattern recognition," *Handbook of Geometric Computing*, pp. 129–167, 2005.

[9] B. McFee and D.P.W. Ellis, "Better beat tracking through robust onset aggregation," in *International conference on acoustics, speech and signal processing*, 2014, ICASSP.

[10] Christopher Harte, *Towards automatic extraction of harmony information from music signals*, Ph.D. thesis, University of London, 2010.

[11] "Reference annotations: The Beatles," 2009, `http://isophonics.net/content/reference-annotations-beatles`.

[12] 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*, 2011, pp. 555–560.

[13] Oriol Nieto and Tristan Jehan, "Convex non-negative matrix factorization for automatic music structure identification," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 2013.

[14] J.S. Downie, "The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research," *Acoustical Science and Technology*, vol. 29, no. 4, pp. 247–255, 2008.

[15] Ron J Weiss and Juan Pablo Bello, "Unsupervised discovery of temporal structure in music," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 6, pp. 1240–1251, 2011.

[16] "2013:structural segmentation," June 2013, `http://www.music-ir.org/mirex/wiki/2013:Structural_Segmentation`.