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 as 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 segment boundaries.

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). Serrà et al. incorporate local historical context by concatenating features from successive frames \( x_t \rightarrow [x_t^T, x_{t-1}^T]^T \) prior to computing the self-similarity matrix. We adopt this strategy, with the distinction that \( x_t \) and \( x_{t-1} \) are beat-synchronous.

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 \( i \)th 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 \( U^T: \|R_{i,:} - R_{j,:}\| = \|U^T R_{i,:} - U^T R_{j,:}\| \).

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

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

Reducing \( L \) to \( d < 2t \) principal components retains the most important factors, and normalizing by \( \sigma_1 \) 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).

![Fig. 1](image)

**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).

2.2. 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 \( X_{:,i} \), 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 \( D \in \mathcal{O}(\log t) \), the cost of clustering is dominated by the \( \mathcal{O}(td) \) cost of computing \( X \).

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
where \( A_W \) and \( A_B \) are the within- and between-class scatter matrices:

\[
A_W := \sum_c \sum_{i:y_i=c} (x_i - \mu_c)(x_i - \mu_c)^T, \quad A_B := \sum_c n_c(\mu_c - \mu)(\mu_c - \mu)^T,
\]

\( \mu \) denotes the mean across all classes, \( \mu_c \) 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 \((A_B, A_W)\) [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:

\[
A_O := \sum_{c \leq C} n_c(\mu_c - \mu_{c+})(\mu_c - \mu_{c+})^T + n_{c+1}(\mu_{c+} - \mu_{c+})(\mu_{c+} - \mu_{c+})^T
\]

\[
\mu_{c+} := \frac{n_c \mu_c + n_{c+1} \mu_{c+1}}{n_c + n_{c+1}}.
\]

Intuitively, \( A_O \) measures the deviation of successive segments \((c, c+1)\) from their mutual centroid \( \mu_{c+} \), which coincides with 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 \( \lambda > 0 \). The OLDA optimization takes the form:

\[
W := \arg\max_W \text{tr} \left( (W^T A_W + \lambda I) W \right)^{-1} W^T A_B W, \quad (3)
\]

which again can be solved efficiently as a generalized eigenvalue problem over the matrix pair \((A_O, A_W + \lambda I)\).

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 \( X \mapsto W^T X \), and then clustered as described in Section 2.2.

\( ^1 \)The same regularization strategy is applied to FDA in Section 4.
4. EVALUATION

All proposed methods are implemented in Python with the librosa package. All signals were downsampled to 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 log-magnitude chroma bins; repetition features are calculated with $2\sqrt{t}$ nearest neighbors under a standardized euclidean metric, median-filtered with a window of width 7, and reduced 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

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 chorus), 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-PLCA [15].

For both datasets, we evaluate the unweighted feature representation (Native), 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 $F_{0.5}$ score (see below) on the training set.

For each method and dataset, we report the precision, recall, and F-measure of boundary retrieval, using either a 0.5-second or 3-second window.

4.2. Results

Table 1 lists the results for the Beatles-ISO and SALAMI-free. SMGA achieves highest performance with a 3-second window on both datasets. The proposed methods achieve higher accuracy for boundary detection at 0.5s resolution, which can be attributed to the increased accuracy afforded by the beat-synchronous and median-filtered repetition features.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>$P_{0.5}$</th>
<th>$R_{0.5}$</th>
<th>$F_{0.5}$</th>
<th>$P_{5}$</th>
<th>$R_{5}$</th>
<th>$F_{5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beatles-ISO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Native</td>
<td>0.278</td>
<td>0.271</td>
<td>0.266</td>
<td>0.560</td>
<td>0.510</td>
<td>0.522</td>
</tr>
<tr>
<td>FDA</td>
<td>0.304</td>
<td>0.310</td>
<td>0.296</td>
<td>0.555</td>
<td>0.536</td>
<td>0.530</td>
</tr>
<tr>
<td>OLDA</td>
<td>0.297</td>
<td>0.320</td>
<td>0.296</td>
<td>0.553</td>
<td>0.550</td>
<td>0.535</td>
</tr>
<tr>
<td>SMGA</td>
<td>0.144</td>
<td>0.169</td>
<td>0.153</td>
<td>0.621</td>
<td>0.728</td>
<td>0.658</td>
</tr>
<tr>
<td>SALAMI-free</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Native</td>
<td>0.233</td>
<td>0.188</td>
<td>0.202</td>
<td>0.492</td>
<td>0.406</td>
<td>0.431</td>
</tr>
<tr>
<td>FDA</td>
<td>0.290</td>
<td>0.222</td>
<td>0.243</td>
<td>0.552</td>
<td>0.429</td>
<td>0.466</td>
</tr>
<tr>
<td>OLDA</td>
<td>0.255</td>
<td>0.248</td>
<td>0.247</td>
<td>0.505</td>
<td>0.483</td>
<td>0.477</td>
</tr>
<tr>
<td>SMGA</td>
<td>0.115</td>
<td>0.178</td>
<td>0.134</td>
<td>0.434</td>
<td>0.666</td>
<td>0.508</td>
</tr>
<tr>
<td>CNMF</td>
<td>0.105</td>
<td>0.133</td>
<td>0.110</td>
<td>0.450</td>
<td>0.543</td>
<td>0.463</td>
</tr>
<tr>
<td>SI-PLCA</td>
<td>0.210</td>
<td>0.102</td>
<td>0.128</td>
<td>0.451</td>
<td>0.228</td>
<td>0.286</td>
</tr>
</tbody>
</table>

FDA and OLDA consistently improve over the baseline (Native), and achieve comparable F-scores. However, the FDA method tends to improve precision at the expense of recall, which indicates a tendency toward under-segmentation. The OLDA method is adapted to only distinguish successive segments, and exhibits better boundary recall.

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.

6. ACKNOWLEDGMENTS

The authors thank Oriol Nieto for discovering an evaluation error in an early draft of this work. The authors acknowledge support from The Andrew W. Mellon Foundation, and NSF grant IIS-1117015.

7. REFERENCES


