THEORY AND EVALUATION OF A BAYESIAN MUSIC STRUCTURE EXTRACTOR

Samer Abdallah, Katy Noland, Mark Sandler
Centre for Digital Music
Queen Mary, University of London
Mile End Road, London E1, UK
samer.abdallah@elec.qmul.ac.uk
katy.noland@elec.qmul.ac.uk
mark.sandler@elec.qmul.ac.uk

Michael Casey, Christophe Rhodes
Centre for Cognition, Computation and Culture
Goldsmiths College, University of London
New Cross Gate, London SE14 6NW, UK
m.casey@gold.ac.uk
c.rhodes@gold.ac.uk

ABSTRACT

We introduce a new model for extracting end points of music structure segments, such as *intro*, *verse*, *chorus*, *break* and so forth, from recorded music. Our methods are applied to the problem of grouping audio features into continuous structural segments with start and end times corresponding as closely as possible to a ground truth of independent human structure judgements. Our work extends previous work on automatic summarization and structure extraction by providing a model for segment end-points posed in a Bayesian framework. Methods to infer parameters to the model using Expectation Maximization and Maximum Likelihood methods are discussed. The model identifies all the segments in a song, not just the chorus or longest segment. We discuss the theory and implementation of the model and evaluate the model in an automatic structure segmentation experiment against a ground truth of human judgements. Our results shows a segment boundary intersection rate break-even point of approximately 80%.

Keywords: structure, segmentation, audio, music, Bayes

1 INTRODUCTION

Methods for automatically segmenting music recordings into structural segments, such as *verse* and *chorus*, have immediate applications in music summarization, song identification, feature segmentation, feature compression and content-based music query systems. In order to evaluate an automatically-generated segmentation, however, we must develop an understanding of both the act of segmentation and the use to which a segmentation will be put.

Studies into automatic music structure extraction assume at least one self-similar region within a work. The definition of self-similarity varies significantly between reported methods, but the concept of a homogeneous region (in some feature space) defined by start and end times is germane to all of the methods that we summarize below.

One factor affecting automatic structure extraction is that the choices of audio features and similarity measure over the space defined by those features crucially determine the attributes of self-similarity, and therefore, the definition of a segment; audio features are generally chosen for specific applications based on their selectivity of well-understood musical attributes such as timbre and harmony, for example. A second factor is the desired distribution of segment lengths. There may be many valid segmentations of a piece of music, distinguished by their different time scales, and possibly also their time point of origin.

Low-level audio feature representations are short duration samples of continuous processes. The integration of samples to form variable-length homogeneous segments is a primary goal of structure extraction. However, the time scale of the segmentation, or the number of segments found, will be sensitive to the scale of the integration over these low-level features. A common problem of many previously proposed systems is a high degree of fragmentation in the discovered segments.

In this paper we discuss previous approaches to audio segmentation, and present a probabilistic model for structure segmentation that attempts to address the problem of segment fragmentation. The model admits accurate estimation of the end-points of all structure segments, not just the 'key' segment or chorus. We evaluate our model against a ground truth of all structure segments within a set of fourteen popular song recordings, and discuss planned extensions to our system.

1.1 Segmentation by Timbre

One way of making structure out of music is to segment based on "the way it sounds" (Aucouturier et al., 2005); that is: to consider the mixture of instrument timbres, chords and pitches that are all encoded in the power spectrum, but in a decorrelated representation such as Mel Frequency Cepstral Coefficients (MFCCs). Features of this form correspond to strong selectivity of wide-band modulation in the source power spectrum. This representation is widely used in speech recognition systems because of the ability to represent formants by a linear combination of cosine basis functions over the log power spectrum.

Foote (1999) proposed the dissimilarity matrix or S-matrix, containing a measure of dissimilarity for all pairs of feature vectors, for music structure analysis using MFCC features. With a feature sample rate of 100Hz this meant that a 3-minute song produced an dissimilarity matrix with dimensions 18000×18000. All of the proposed operations in sequence based methods utilize this extremely large, dense data object, related to the recurrence plot discussed in Eckmann et al. (1987); for instance, Foote proposed that the chorus should be labelled as the longest self-similar segment, using cosine distance measures and MFCC features.

Logan and Chu (2000) proposed a method for summarization, also using MFCCs, employing both Hidden-Markov Models (HMMs) and threshold-based clustering methods, grouping features into key song segments. Peeters et al. (2002) propose a multi-pass clustering approach that uses both k-means and HMM-based clustering using multi-scale MFCC features. However, these studies provide no measure of performance for all segments in a song.

1.2 Harmony

The timbral approach has the potential to be implicitly invariant to harmonic shifts, though this depends on the degree of selectivity for purely timbral content. Recent studies, however, have posed the structure extraction problem in terms of features based on harmonic similarity. Wakefield (1999) proposed chromagram features that represent the distribution of power spectrum energies among the twelve equal-temperament pitch classes based on A440, providing invariance to timbral changes in repeated segments.

One desirable property of harmonic features is the possibility of implementing explicit transpositional invariance. Goto (2003) describes a system called *RefraiD* that locates repeated structure segments independent of transposition. The *RefraiD* system is able to track a chorus, for example, even if it modulates up a sequence of semitone key changes. In this study, the problem of chorus extraction was divided into four stages: acoustic features and similarity measure; repetition judgement criterion; estimating both ends of repeated sections; and detecting modulated repetition. This was the first work to explore extracting structure segments that were not only *chorus* but that corresponded to *verse* and *intro* as well. The results for chorus detection were reported as accurate for 80 of 100 songs. However, the quality of the segmentation for non-chorus segments was not evaluated in that study.

Dannenberg and Hu (2002) also describe a system that used agglomerative clustering with chroma-based features for music structure analysis of a small set of Jazz and Classical pieces. They do not report an evaluation of the methods over a corpus.

1.3 Rhythm and Pitch

Symbolic approaches to structure analysis attempt to identify the repeated thematic material in string-based music representations. Whilst these methods show much promise in identifying structure from score information, they are not well adapted for use in audio structure analysis, largely due to the addition of significant uncertainty in audio representations.

There has recently been some work on combined audio and symbolic representations, attempting to unify the different views of similarity. Maddage et al. (2004) describe a system in which a partial transcription is used to make decisions about structure, integrating beat tracking, rhythm extraction, chord detection and melodic similarity in a heuristic framework for detecting all segments in a song. They also propose using octave-scale rather than Mel-frequency scale cepstral coefficients as pitch-oriented representation. The authors report 100% accuracy for detecting instrumental sections in songs, and report results for detection and labelling of verse, chorus, bridge, intro and outro sections. Similarly, Lu et al. (2004) describe an HMM-based approach to segmentation that used a $\frac{1}{12}$th-octave constant-Q filterbank for pitch selectivity in addition to MFCC features. They report improved performance in segmentation for the constant-Q transform when used with MFCC over use of MFCC alone. Both of these methods used an S-matrix approach with an exhaustive search to find the best fit segment boundaries to a given objective function.

2 SEGMENTATION METHODS

2.1 Overview

To perform a segmentation, by which we mean assigning a label to sections of a piece, we start by preprocessing the audio into a Markov state sequence modelling the short-time dynamics (section 2.2. These state labels are then clustered into segments by various methods described in sections 2.3 and 2.4.

2.2 Audio preprocessing

The tool-chain we work with takes as its input mono audio in WAVE format (IBM, 1991) and performs various 'preprocessing' stages[1] to convert this into a suitable form for the segmenter. The first stage is to perform a short-time (windowed) Fourier Transform to obtain a representation of the frequency spectrum at given times from the beginning of the track. The resulting linear-frequency power spectrum is collected into logarithmically-spaced bins, and expressed in decibels, in a manner similar to the first stage in the construction of a log-frequency cepstrum, such as those often used in speech analysis (Rabiner and Juang, 1993).

Rather than performing a Fourier transform to obtain the first few cepstral coefficients, we use a data-driven algorithm to find the best way to reduce the dimensionality of the data: a principal components analysis (PCA) identifies the main directions of variation in the log-frequency log-spectra and hence the best (in a least-squares sense)

[1] These preprocessing stages correspond closely to descriptors `AudioSpectrumEnvelopeD`, `AudioSpectrumProjectionD`, `SoundModelDS` and `SoundModelStatePathD`, defined in the MPEG-7 standard (Casey, 2001; ISO, 2002).

low-dimensional approximation to the data. The spectra are projected into this N-dimensional principal subspace, and used to train an M-state Hidden Markov Model. The Viterbi algorithm yields the most probable state path given the data and the trained HMM, and from this state sequence, sequence of short-term state occupancy histograms (i.e. with M bins, one for each state) are constructed.

2.3 Pairwise clustering

Results from the above audio processing steps inhabit an M-dimensional space which is not self-evidently Euclidean; clustering methods based on Euclidean feature values are not trivially applicable. One way to proceed is to define empirical dissimilarity measures between observed windowed state histograms with reasonable properties: histograms with the same distribution should be maximally similar, while those with no overlap should be maximally dissimilar.

One such is the cosine dissimilarity measure, expressed for l^2-normalized histograms $\mathbf{x}$ and $\mathbf{x}'$ as $d_c(\mathbf{x}, \mathbf{x}') = \cos^{-1}(\mathbf{x} \cdot \mathbf{x}')$; this dissimilarity measure naturally interprets the feature vector components as of equal importance with no data-driven preferred orientation.

Another distance measure is a symmetrization of the Kullback-Leibler divergence, based on the interpretation of the histograms as samples from a probability distribution. With l^1-normalized histograms $\mathbf{x}$, $\mathbf{x}'$, we have $d_{\text{kl}}(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{M} \left[x_i \log\left(x_i q_i^{-1}\right) + x_i' \log\left(x_i' q_i^{-1}\right) \right]$ where $q_i = \frac{1}{2}(x_i + x_i')$. This can be interpreted as the sum of the KL divergences from either histogram to their mutual average.

These pairwise distances are then used in assigning frames to clusters; we make iterative probabilistic assignments of frames to clusters to minimize a cost function. We refer the reader to Hofmann and Buhmann (1997) for details, mentioning only that in order to perform the optimisation we work with the mean-field approximation for interactions between assignments, and use the Expectation Maximization algorithm (Dempster et al., 1977) to anneal towards a minimum in the cost function.

2.4 Histogram clustering

Since the data we wish to cluster can be interpreted as observation counts over some discrete feature space, we may, following Puzicha et al. (1999), consider each cluster to have a probability distribution over the feature space. The observed histograms are then modelled as the result of drawing a sample from one of these distributions. This leads quite naturally to a probabilistic latent variable model with a well defined likelihood function which we can optimize with respect to its parameters.

The discrete distributions associated with the K underlying clusters are parameterised by an $M \times K$ matrix $\mathbf{A}$, such that A_{jk} is the probability of observing the jth HMM state in while in the regime modelled by the kth cluster. If $\mathbf{C} \in (1..K)^L$ is the sequence of cluster assignments for a given sequence of histograms $\mathbf{X} \in \mathbb{N}^{M \times L}$, then the overall log-likelihood of the model reduces to

$$\mathcal{H}_h = \sum_{i=1}^{L} \sum_{j=1}^{M} \sum_{k=1}^{K} \delta(k, C_i) X_{ji} \log \frac{X_{ji}}{A_{jk}} \qquad (1)$$

This overall system can be compared with a form of k-means clustering, though with a strongly non-Euclidean distance measure between observations: the maximum *a posteriori* estimate for $P(y|c)$, which generalizes the centroid condition of k-means clustering, works in the space of feature distributions and not feature values.

This cost function is optimised using a form of deterministic annealing as described by Puzicha et al. (1999), which yield the maximum *a posteriori* estimate for the latent variables $\mathbf{C}$ and the per-cluster histograms in $\mathbf{A}$.

3 EXPERIMENTS

We performed segmentations using the above-described methods on 14 popular music songs from Sony's catalogue, which had been down-sampled to 11kHz mono before being distributed to the MPEG-7 working group. The FFT frame length was set at 600ms (with a hop size of 200ms) giving an FFT window of 8192 samples. The discrete Fourier transforms were then clustered into bins with a resolution of $\frac{1}{8}$-octave, and the resulting data represented in terms of its envelope and 20 principal components.

We further fixed the window size for state histograms at 15 states, giving a temporal support of 3s to each histogram (though the histogram window is in fact applied to all possible positions, preserving the 200ms resolution). The segmentations were performed using HMMs with 10, 20, 40 and 80 states, and into a numbers between 2 and 10 of segment types.

A sample segmentation, along with some of the intermediate results, is presented in figure 1.

4 EVALUATION

In order to evaluate the segmentations, they were compared against a ground truth consisting of annotations made by an expert listener, giving, for each ground truth segment, a start time in seconds, an end time in seconds and a label.

To make the comparison it is necessary to map the boundaries between clusters back to the original continuous timeline, on which the ground truth annotations are based. Bearing in mind that sequence of short-term histograms is defined on a discrete timeline which is itself derived via two framing operations from the original discrete time signal, this is not a trivial operation. Depending on how the labelled moments are interpreted, the boundary between two segments (essentially the 'gap' between two discrete time moments) could be mapped back to one of several points or intervals on the continuous timeline. We shall, for the time being, choose the simplest option and map the gap between two discrete time moments back to to middle of the overlap between their respective continuous time intervals, which, at 15 HMM states per histogram, are approximately 3s long.

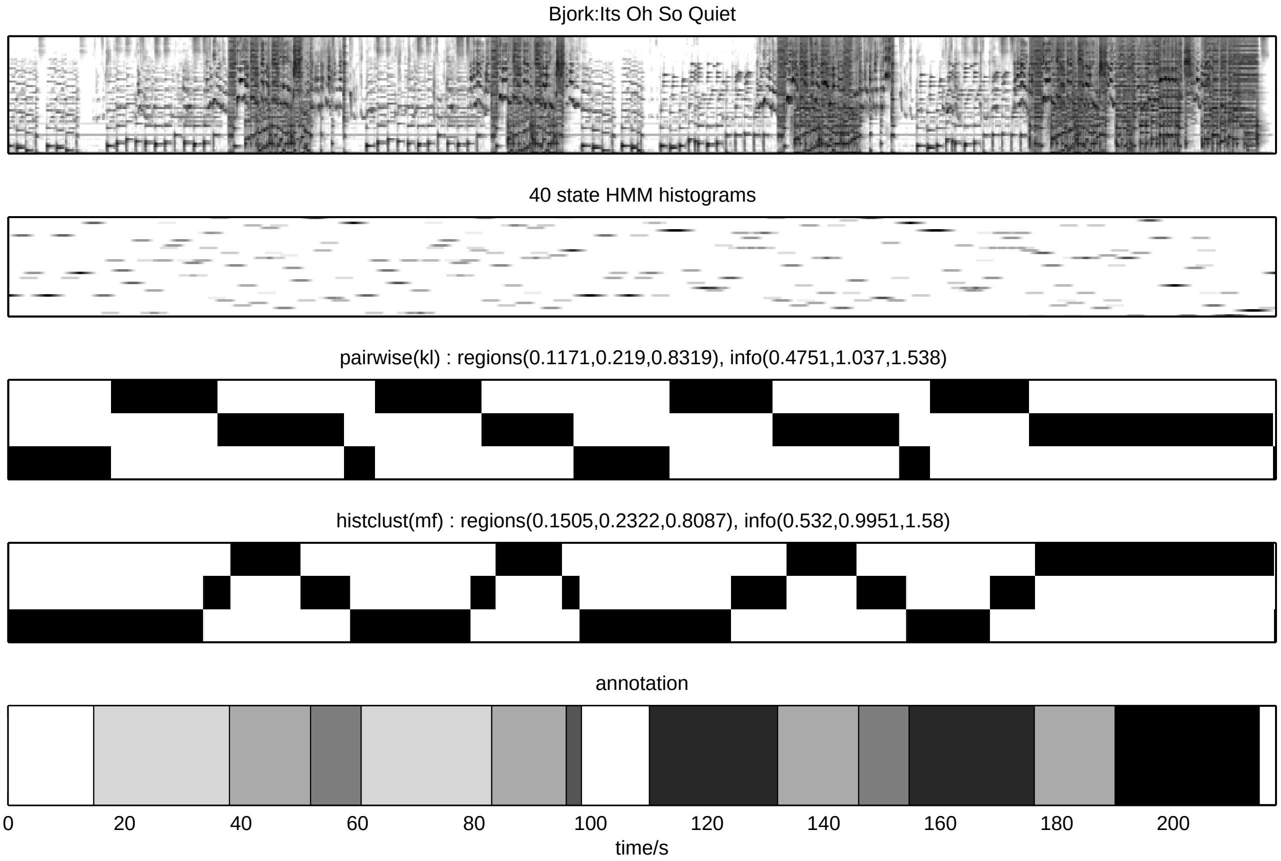

Figure 1: A segmentation of a sample from the test set, comparing the results of dyadic clustering (using the symmetrized Kullback-Leibler distance) and the histogram clustering algorithm. The 'ground truth' annotations are displayed as different shades of grey for the different segment labels.

Having found times for the detected segment boundaries, we adapted the segmentation evaluation measure of Huang and Dom (1995). Considering the ground truth G as a sequence of segments S_G^i, and the measurement M likewise segments S_M^j, we compute a directional Hamming distance d_{GM} by finding for each S_M^i the segment S_G^j with the maximum overlap, and then summing the difference,

$$d_{GM} = \sum_{S_M^i} \sum_{S_G^k \neq S_G^j} |S_M^i \cap S_G^k| \qquad (2)$$

where $|\cdot|$ denotes the length of a segment. We normalise d_{GM} by the track length L to give a measure of the missed boundaries $m = \frac{d_{GM}}{L}$. Similarly, we compute d_{MG}, the inverse directional Hamming distance, and a similar normalised measure $f = \frac{d_{MG}}{L}$ of the segment fragmentation. Note that these measures consider only the time intervals occupied by each segment, not the relationship between the ground truth labels are the automatically assigned clusters. Plots of f and m against the number of clusters for our corpus are presented in figures 2 and 3.

An alternative information-theoretic measure was also investigated in order to assess the how well the clusters reflect the original segment labels. This involves 'rendering' the ground-truth segmentation into a discrete time sequence of numeric labels $\mathbf{C}_0$, using the same discrete timebase as the sequence to be assessed, $\mathbf{C}_1$. The two sequences are then compared by computing the conditional entropies $H(\mathbf{C}_1|\mathbf{C}_0)$ and $H(\mathbf{C}_0|\mathbf{C}_1)$ and the mutual information $I(\mathbf{C}_0, \mathbf{C}_1)$ by considering the joint probability distribution over labels.

The mutual information $I(\mathbf{C}_0, \mathbf{C}_1)$ measures the information in the cluster assignments about the ground truth segment label, and is maximal when each segment type maps to one and only one cluster. In this case both $H(\mathbf{C}_1|\mathbf{C}_0)$ and $H(\mathbf{C}_0|\mathbf{C}_1)$ will be zero. $H(\mathbf{C}_0|\mathbf{C}_1)$ measures the amount of ground-truth information 'missing' from the cluster assignments, while $H(\mathbf{C}_1|\mathbf{C}_0)$ measures the amount of 'spurious' information in the cluster assignments, e.g. when several clusters represent one segment type. We plot the mutual information for our segmentation methods in figure 4.

5 CONCLUSIONS

Firstly, it is clear from the individual results (one of which was presented in section 3) that the approach we have taken in this paper, to a large extent independent of the details of the particular segmentation algorithm, has met with a degree of success. While no segmentation produced by our algorithm was perfect, some (represented in the top right corner of figure 5, which is analogous to a precision-recall graph for our evaluation metrics) are close to the ideal of the expert's segmentation.

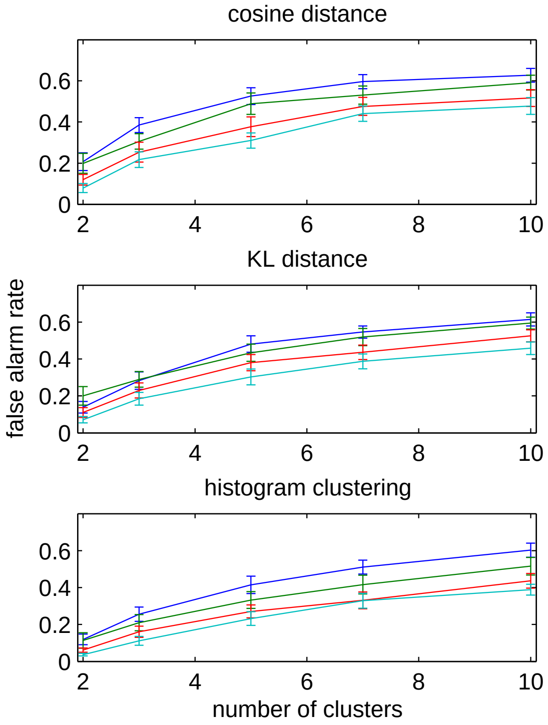

Figure 2: Rate of false detection f for all segmentation methods aggregated over our corpus. The four curves are for HMMs with 10,20,40 and 80 states; there is no strongly statistically significant difference between them.

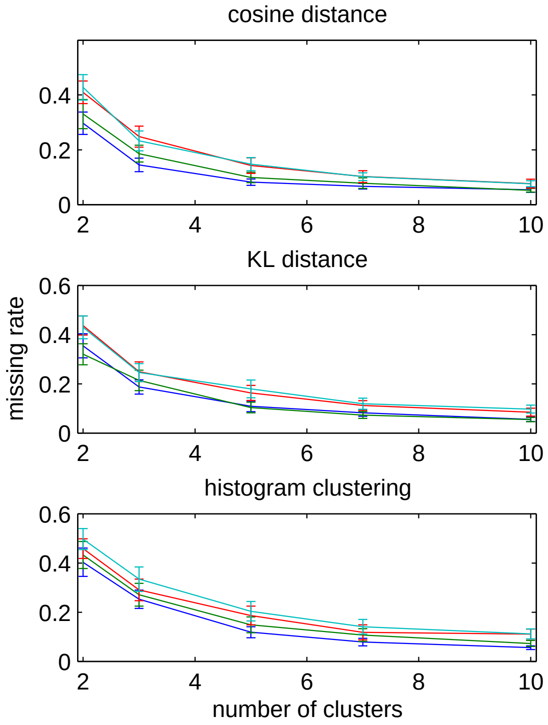

Figure 3: Rate of true negative failure m for all segmentation methods aggregated over our corpus. As in figure 2, the four curves display the data for HMMs with different numbers of states.

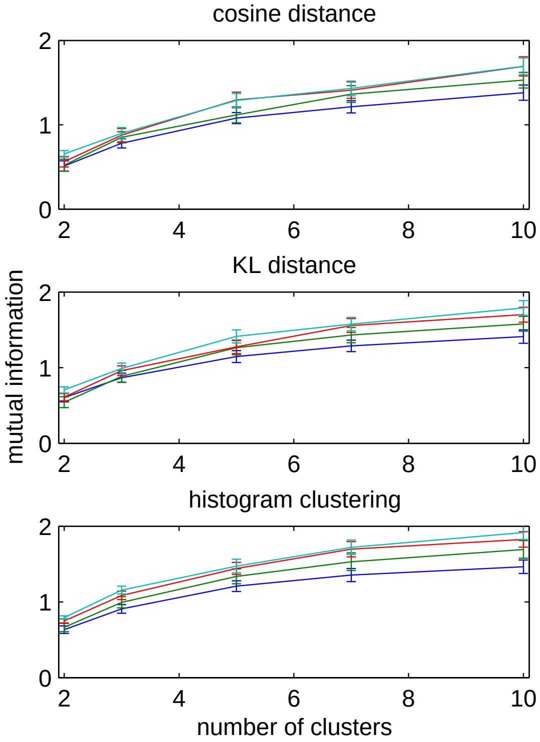

Figure 4: Mutual Information (in bits) between ground truth and machine segmentation for our segmentation methods.

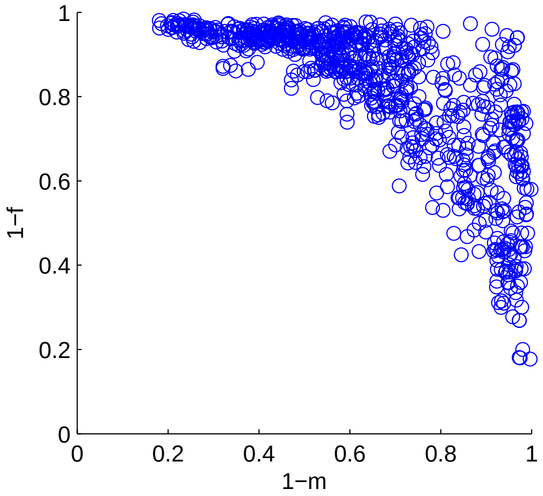

Figure 5: Values of $1 - f$, corresponding loosely to precision, plotted against values of $1 - m$, analogous to recall, over all songs and segmentation methods presented.

We should note that the expert's segmentation should not be taken as the Platonic truth: equally valid segmentations, depending on the application, can be formed at greatly different timescales; in addition, in real music there is often a degree of ambiguity as to the exact point of transition between one segment and the next: an ambiguity which was not reflected in the expert's judgement, as the human segmentation was declared as precise to the frame level.

Having noted the caveat of the previous paragraph, it is nevertheless clear that the rate of true negative failure decreases as the number of clusters increases, while the false positive detection increases. This is simply explained by the fact that, beyond a certain point, increasing the number of clusters will cause oversegmentation and fragmentation: a segment which is semantically uniform will be divided between two closely-related states, each given a different label, and the cluster assignment will oscillate between the two. Requesting a greater number of clusters from the algorithm, then, has the effect of decreasing the average length of a machine-labelled segment, which means that a machine-labelled boundary is likely to be close to a boundary labelled by the expert, but also that there are likely to be more machine labelled boundaries far away from those of the expert.

We have found that excessive fragmentation of labelled sections can be avoided in part with a reasonable choice of parameters. However, there is still scope for improvement, and two areas for investigation present themselves: firstly, fragmentation can be in large part solved by cluster aggregation based on an information-theoretic treatment with Occam's razor. Secondly, and perhaps more intruigingly, the careful choice of a prior on cluster duration, tailored to the application, could permit relaxing the initial segmentation to a smoother, less fragmented one.

In a bid to keep the parameter space tractable for this investigation, we have not discussed variations in the audio 'preprocessing' chain. In addition to the obvious parameters which could be varied, such as FFT hop size and band resolution, the effects of considering a feature representation such as a chromagram, in place of or in addition to the cepstrum that we have used, warrant investigation. We leave this for further work, anticipating that the performance of the system will improve with the addition of harmonically-selective features to the timbral cepstrum.

ACKNOWLEDGEMENTS

This research was supported by EPSRC grant GR/S84750/01 (Hierarchical Segmentation and Semantic Markup of Musical Signals).

REFERENCES

J.-J. Aucouturier, F. Pachet, and M. Sandler. The way it sounds : Timbre models for analysis and retrieval of polyphonic music signals. *IEEE Transactions of Multimedia*, 2005.

M. Casey. MPEG-7 sound-recognition tools. *IEEE Trans. Circuits Syst. Video Techn.*, 11(6):737–747, 2001.

R. Dannenberg and N. Hu. Discovering musical structure in audio recordings. In *Music and Artifical Intelligence: Second International Conference*, Edinburgh, 2002.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.

J.-P. Eckmann, S. O. Kamphorst, and D. Ruelle. Recurrence plots of dynamical systems. *Europhysics Letters*, 5:973–977, 1987.

J. Foote. Visualizing music and audio using self-similarity. In *ACM Multimedia (1)*, pages 77–80, 1999.

M. Goto. A chorus-section detecting method for musical audio signals. In *Proc. ICASSP*, volume V, pages 437–440, 2003.

T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1), 1997.

Q. Huang and B. Dom. Quantitative methods of evaluating image segmentation. In *Proc. IEEE Intl. Conf. on Image Processing (ICIP'95)*, 1995.

Multimedia Programming Interface and Data Specifications 1.0. IBM Corporation and Microsoft Corporation, August 1991.

Information Technology – Multimedia Content Description Interface – Part 4: Audio. ISO, 2002. 15938-4.

B. Logan and S. Chu. Music summarization using key phrases. In *International Conference on Acoustics, Speech and Signal Processing*, 2000.

L. Lu, M. Wang, and H. Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, October 2004.

N. Maddage, X. Changsheng, M. Kankanhalli, and X. Shao. Content-based music structure analysis with applications to music semantics understanding. In *6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, October 2004.

G. Peeters, A. L. Burthe, and X. Rodet. Toward automatic music audio summary generation from signal analysis. In *International Symposium on Music Information Retrieval*, 2002.

J. Puzicha, T. Hofmann, and J. M. Buhmann. Histogram clustering for unsupervised image segmentation. *Proceedings of CVPR '99*, 1999.

L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Signal Processing Series. Prentice Hall, Englewood Cliffs, NJ, 1993.

G. H. Wakefield. Mathematical representation of joint time-chroma distributions. In *Advanced Signal Processing Algorithms, Architectures, and Implementations*, volume 3807, IX, pages 637–645. SPIE, 1999.