A DYNAMIC PROGRAMMING APPROACH TO AUDIO SEGMENTATION AND SPEECH / MUSIC DISCRIMINATION

Michael M. Goodwin and Jean Laroche

Creative Advanced Technology Center
Scotts Valley, CA
mgoodwin,jeanl@atc.creative.com

ABSTRACT

We consider the problem of segmenting an audio signal into characteristic regions based on feature-set similarities. In the proposed approach, a feature-space representation of the signal is generated; sequences of these feature-space samples are then aggregated into clusters corresponding to distinct signal regions. The algorithm consists of using linear discriminant analysis (LDA) to condition the feature space and dynamic programming (DP) to identify data clusters. In this paper, we consider the design of the dynamic program cost functions; we are able to derive effective cost functions without relying on significant prior information about the structure of the expected data clusters. We demonstrate the application of the LDA-DP segmentation algorithm to speech / music discrimination; experimental results are given and discussed.

1. INTRODUCTION

Segmentation of audio signals into meaningful regions is an essential component of many applications, for instance speech / music discrimination for broadcast transcription, transient detection for window-switching audio coders, audio thumbnailing, and demarcation of songs in continuous streams for database creation or smart transport. Such applications rely on a basic signal understanding provided by automatic segmentation.

Segmentation approaches described in the literature generally represent the signal as a sequence of features in a meaningful feature space and then attempt to identify points-of-change in the feature sequence using statistical models or various distance metrics [1, 2, 3]. In this paper we present a novel approach similarly based on intuitive signal features but which does not rely on the heuristic hard-decision thresholding typical of other methods.

In the proposed algorithm, the segmentation task is interpreted as a feature-space clustering problem. First, the feature-space clustering behavior is improved by the use of linear discriminant analysis trained on representative examples; then, a dynamic program is used to derive robust robust segmentation points – which correspond to cluster transitions in the problem framework. We describe the structure of the dynamic program and derive cost functions which lead to robust clustering.

The LDA-DP segmentation algorithm was initially described by the authors in [4]. Here, we elaborate on the design of the dynamic program and focus on the specific application of the LDA-DP method to the task of speech / music discrimination (SMD). We illustrate how the LDA conditions the feature space for clustering but does not serve as an effective stand-alone clustering solution, and we demonstrate how the DP in the conditioned feature space is able to robustly detect points of change in the SMD task.

2. FEATURE SPACE

An audio signal can be represented in a feature space by carrying out a sliding-window analysis to extract feature sets on a frame-to-frame basis. In such a scheme, each hop of the window $w[n]$ yields a new set of features; for the i-th frame:

$$w[n]x[n+iL] \rightarrow \boxed{\text{Feature analysis}} \rightarrow f_i\{x\} \qquad (1)$$

The output of the feature analysis block is the feature vector $f_i\{x\}$, which will be treated as a column vector f_i. Examples of features relevant to the audio segmentation task include zero-crossing rate, subband energies, spectral envelope, centroid, tilt, and flux, and so on [2]; similar features are used in music information retrieval, audio fingerprinting, and other content-based applications [5]. The selection of features is a key aspect in the performance of such systems; for instance, if the feature set used in fingerprinting does not exhibit sufficient differences for different songs, the fingerprinting will not perform robustly. One of the design goals in the proposed segmentation scheme is to avoid the pitfalls of feature selection; our approach is to incorporate a wide array of features and apply prior training to weight those features appropriately for the task.

2.1. Segmentation by feature-space distance

The sequence of feature vectors f_i provides a feature-space representation of the input signal from which a variety of similarity (dissimilarity) metrics can be computed for successive feature vectors. A typical metric is the vector difference norm

$$d_{ij} = (f_i - f_j)^H D^H D (f_i - f_j), \qquad (2)$$

where $D^H D$ is the identity matrix for the Euclidean distance, the inverse covariance matrix of the feature set for the Mahalanobis distance, or some other feature weighting specific to the particular distance measure. The sequence of differences between successive feature vectors is then an example of a *novelty function* which attempts to quantify the extent of change in the audio signal between adjacent frames [3]. Feature-based segmentation schemes typically determine segment boundaries by finding peaks in the novelty function; these are taken to indicate points of change in the audio signal, *i.e.* there is a change if successive features f_i and f_{i+1} are deemed dissimilar enough [2, 3].

In novelty-function segmentation approaches, a heuristic threshold is used to make the decision as to whether successive frames are substantially dissimilar to indicate a segmentation boundary. As illustrated in [4], this is problematic in that typical novelty functions tend to exhibit peaking not only in the vicinity of segment boundaries but also within segments. As a result, peak-picking can readily lead to incorrect segment boundary determinations.

2.2. Segmentation as clustering

This spurious peaks in novelty functions can be understood by considering segmentation as a clustering problem: clusters of data corresponding to distinct segments are observed sequentially; segmentation is equivalent to finding the transitions between clusters. The distance between successive features in the same cluster (segment) can rival or exceed the distances between features in different clusters, so novelty peak-picking can indicate faulty mid-cluster segmentation boundaries.

The extraneous novelty peaks tend to be most pronounced for the Euclidean distance since the distance measure be dominated by features having large variance within a cluster but little value in discriminating between clusters. As an alternative, Mahalanobis feature weighting can be used; this essentially normalizes the contribution of each feature, which is reasonable if no prior information is available about the relative discriminatory value of the various features, but it may tend to devalue the most discriminatory features. If representative examples of the desired clusters are available *a priori*, linear discriminant analysis (LDA) can be carried out on those examples to yield a feature-space transformation which accentuates the discriminatory features. Using the LDA matrix in the weighted distance of Eq. (2) yields a novelty function which tends to display less peaking within clusters and stronger peaking at the cluster transitions. This improvement occurs because the LDA is trained to sphere the data classes and separate the class means [6]; the improvement is notable for clustering of new data if the training set is representative of the desired clusters.

A further benefit of LDA is that it addresses the feature selection problem. The feature samples in the LDA training classes can contain an arbitrarily large number of features. The LDA training determines the best linear combination of these features to separate the data classes while projecting the data onto a subspace with dimension equal to one less than the number of training classes (or fewer). LDA thus results in dimension reduction as well as automatic management of the relevance of the raw signal features.

While the use of LDA-weighted distance does improve the problem of spurious peaking, such peaks still occur. The reason for this is a basic shortcoming of the novelty function: it is designed to identify local changes between samples; global changes between groups of samples, however, are of greater importance to the segmentation task. LDA does not resolve this issue entirely since it is not at all designed to enhance inter-sample novelty but moreso inter-cluster novelty. The spurious peaks arise when two successive samples within a single cluster are further apart than successive samples in different clusters, which is a common occurrence in tightly packed feature spaces – even after a clustering transformation such as LDA. What is needed instead of a local novelty measure, then, is a detection of global signal trends. In the next section, we describe a dynamic program which essentially looks for the means of subsequences and identifies segmentation boundaries when the samples start to aggregate around a new mean.

3. DESIGN OF A DYNAMIC PROGRAM FOR IDENTIFYING SEQUENTIAL CLUSTERS

The segmentation problem differs from standard clustering problems in that the clusters arrive sequentially. In this section, we discuss the structure and design of a dynamic program which takes into account the sequential nature of the feature clustering in the segmentation framework. We describe how to choose cost functions for the DP such that it detects transitions between successive

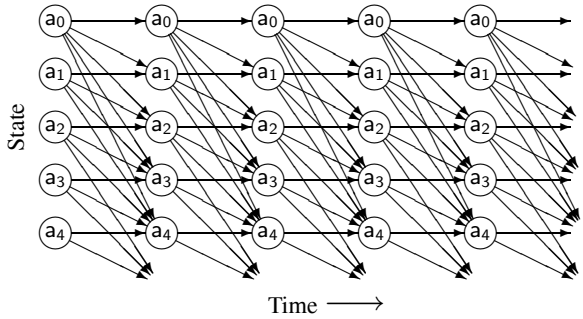

Fig. 1. Partial state transition diagram of the dynamic program for feature-space clustering. The label corresponds to the feature vector associated with the state; state S_{ij} has feature vector a_i.

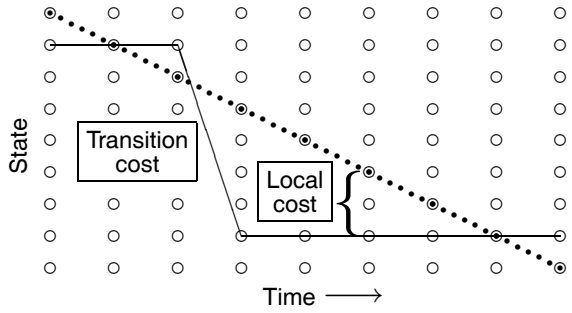

Fig. 2. In the dynamic program for feature-space clustering, the diagonal (dotted) is the nominal feature path. A candidate cluster path with one transition is shown; there is a transition cost as well as a local cost for being in any state that is not on the nominal path.

clusters; the optimal path in the DP follows the general trend of the feature aggregation and is robust to feature-space outliers and spurious peaks in the intra-cluster distance.

3.1. Structure of the dynamic program

Given the LDA-transformed feature sequence $\{a_i\}$, which will exhibit better clustering than the raw data $\{f_i\}$, dynamic programming can be used to find cluster transitions. Assuming there are N feature sets in the sequence, an $N \times N$ state machine such as that in Fig. 1 is constructed. For each time frame j, there are N candidate states; letting i be the vertical state index, each state S_{ij} is associated to the feature vector a_i as shown in the figure.

The diagonal path of the state transition diagram in Fig. 1 corresponds to the nominal feature-space trajectory of the signal: at time j, the *nominal path* is in state j, whose state vector is a_j, namely the actual features generated by the signal at that time. The nominal feature path is depicted in Fig. 2 along with a candidate *cluster path*; the objective of the DP will be to find such a path which indicates cluster transitions. As shown, the cluster path is a stepwise traversal of the state transition diagram. Each plateau in the path corresponds to a cluster and each step is a transition between clusters; the feature vector for a cluster plateau is the characteristic feature set for that cluster. More specifically, then, the objective of the DP is to find this characteristic feature set for the nominal local sequence of feature vectors and then to transition when this characteristic set is no longer a good representative of the nominal features. This objective can be achieved by judicious design of the DP cost functions.

3.2. Cost function design

Recall that DP is able to find a path through the state diagram which optimizes some specified cost function which can be composed of a local cost for each state as well as costs for transitions between states. Denoting a path through the DP as $p[j]$ and the nominal path as $n[j] = S_{jj}$, we can express the optimal path $r[j]$ as

$$r[j] = \arg\min_{p[j]} \sum_j \alpha L\left(p[j], n[j]\right) + \beta T\left(p[j], p[j-1]\right) \quad (3)$$

$$= \arg\min_{p[j]} \{\alpha C_L(p[j]) + \beta C_T(p[j])\} \quad (4)$$

where L and T are the local and transition cost functions, α and β are relative weights for the costs, and C_L and C_T are aggregate totals for the components costs for a given path. The objective of the dynamic program design is to choose the cost functions L and T such that the resulting optimal path indicates a trajectory through clusters which are representative of the nominal feature-space trajectory. We consider these component costs in turn.

The local cost should reflect how reasonable it is to be in state i at time j. Recall that we want the final trajectory to be composed of horizontal segments separated by transitions. In any horizontal segment of this desired cluster path, we remain in the same state i_0: the local cost of state i_0 at time j ($S_{i_0 j}$) should be small if the feature vector measured in the signal at time j (the one associated to the nominal state S_{jj}) is similar to the feature vector associated to state i_0, and high otherwise. This ensures that along a horizontal segment the successive feature vectors extracted from the signal are similar to the feature vector a_{i_0} that represents that segment. An intuitive choice for the cost function is the Euclidean distance $||a_i - a_j||$ or some weighted distance $(a_i - a_j)^H \Phi (a_i - a_j)$ so that the cost of being in state S_{ij} is the distance between the feature vectors of that state and of the nominal diagonal state S_{jj} for that time index. The states between which the local cost distance is measured are indicated in Fig. 2.

The aggregate local cost for a candidate path is the sum of the local costs for the states in the path. For the Euclidean or any similar distance measure, this is clearly zero for the nominal diagonal path. Assuming for the moment, however, that the transition cost is infinite such that a horizontal path must be chosen, we now show that if we adopt this choice of weighted distance, minimizing the total cost results in a horizontal path characterized by an appropriate representative feature vector. Considering a set of N feature vectors and letting a_m be the single feature vector of the chosen horizontal path, the aggregate local cost for the path is

$$C_L = \sum_{j=0}^{N-1} L(a_m, a_j) = \sum_{j=0}^{N-1} (a_m - a_j)^H \Phi (a_m - a_j) \quad (5)$$

which is minimized if a_m is the mean of the set; a_m must however be chosen from the sample set. To find the best choice, we write:

$$C_L = \sum_{j=0}^{N-1} (a_m - \bar{a} + \bar{a} - a_j)^H \Phi (a_m - \bar{a} + \bar{a} - a_j) \quad (6)$$

$$= N(a_m - \bar{a})^H \Phi (a_m - \bar{a}) + \sum_{j=0}^{N-1} (a_j - \bar{a})^H \Phi (a_j - \bar{a}) \quad (7)$$

where the cross-terms in the expansion of Eq. (6) cancel since $\bar{a}$ is the set mean. Noting that the second term in Eq. (7) is not dependent on a_m, we see from the first term that the optimal choice a_m is the set member closest to the mean. Thus, the optimal horizontal path is the path which stays in the state whose feature vector is closest to the mean of the set. In the clustering framework, this feature is the closest member of the cluster to the cluster mean and is the optimal choice to be a representative of the cluster.

The transition cost can be formulated by considering several constraints. First, a high cost should be associated to switching from state i to state j if the corresponding feature vectors are similar; however, there should be zero cost for a transition from i to i (since we are looking for horizontal paths). Conversely, the cost should be small for a transition between very dissimilar feature vectors (so real transitions in the audio are not missed). An intuitive choice for the transition cost between two states is then the inverse of the Euclidian or some weighted distance between the corresponding feature vectors; a constant cost can also be added for any non-horizontal transition to further favor clustering into horizontal segments. As an aside, if we consider the two-cluster case where a path with one transition is desired instead of a strictly horizontal path, we find that the transition cost between the two cluster means should be upper-bounded so that it would be outweighed by the local cost of just staying in a horizontal path:

$$T(m_0, m_1) < \frac{N_0}{2} (m_0 - m_1)^H \Phi (m_0 - m_1), \quad (8)$$

where N_0 is the cluster size and m_0 and m_1 are the cluster means. Some prior knowledge about the structure of the data clusters (such as the size and mean spacing) can thus be helpful in scaling the transition cost appropriately.

We can furthermore incorporate a transition bias cost which depends not on the source and target state feature vectors but rather on their vertical state indices, namely their nominal time indices in the feature vector sequence. For the segmentation task as described, only horizontal or downward transitions are allowed in the DP; for some signal analysis applications, though, it could be useful to include upward transitions in the state diagram and then apply a bias cost to either encourage or discourage revisitation of past clusters. Also, we can impose a lower bound on the size of identified clusters by discouraging or disallowing jumps to indices nearby in the nominal time sequence.

If these local and transition cost functions are used in the DP, the LDA-DP segmentation algorithm robustly identifies segment boundaries. The performance of the LDA-DP segmentation by clustering was demonstrated in [4] for some general audio segmentation tasks; audio examples are available online [7].

4. SPEECH / MUSIC SEGMENTATION AND DISCRIMINATION

There are many audio content management scenarios which would benefit from the ability of some front-end processing to robustly distinguish speech from music. For instance, automatic transcription of broadcast news calls for segmentation of the actual news from musical tag-lines and the like. Another application calling for speech/music discrimination (SMD) is smart transport within a broadcast radio stream; given sufficient buffering, boundaries between songs and commercials or DJ talk could be identified and then used as markers for replay or skip-ahead transport controls.

A variety of approaches to SMD have been presented in the literature. Generally, these rely on a hypothesis testing framework: for an observation of signal features, choose the hypothesis (speech or music) which is most likely to be correct. In short, given

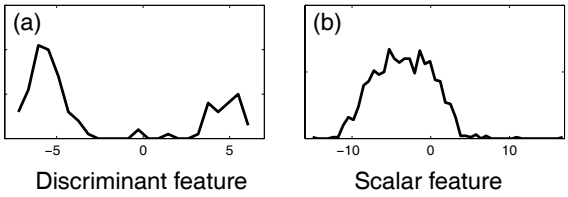

Fig. 3. The distribution of the scalar SMD feature for (a) the training set and (b) a radio broadcast stream.

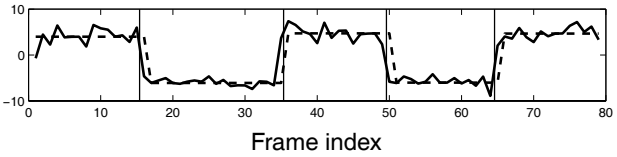

Fig. 4. Plot of the scalar LDA feature (solid) and the cluster features derived by the dynamic program (dashed) for a stream of material from the training set. The vertical lines indicate the actual transitions.

models of speech and music statistics, a new signal feature is classified as either speech or music based on maximum likelihood or some penalized likelihood function such as the Akaike Information Criterion or Bayesian Information Criterion [8]; in some systems, these methods look at a sequence of feature vectors and evaluate the likelihood that a change in the generating model (speech or music) occurred within that sequence.

4.1. Application of the LDA-DP algorithm to SMD

The LDA-DP approach can be readily tailored to SMD. First, the LDA is trained on samples of speech and music. Since there are two data classes, the result of the LDA transformation is a one-dimensional projection of the raw feature data; the LDA matrix A is an $1 \times N$ vector and each LDA feature $a_i = Af_i$ is simply a scalar. Scalar features for the speech and music classes used in the LDA training are depicted in Figure 3(a), where the feature values have been binned to show rough distributions for each class. Here, it is clear that the LDA separates the prior classes well enough that new data similar to the training set could be accurately classified using a simple thresholding decision. When the LDA projection is carried out on a signal not from the training set as in Figure 3(b), the clusters are not necessarily well separated and a decision threshold would lead to major errors. In this case, the broadcast radio stream segments are not at all similar to the training set [7]; in less degenerate cases, better clustering would be apparent, but this example was chosen to illustrate the effectiveness of the DP.

Figure 4 shows the operation of the LDA-DP on material drawn from the training set. The solid line depicts the nominal scalar LDA features a_i, which display significant discrimination between the speech and music segments. The dashed line is the scalar feature sequence for the optimal cluster path derived by the DP; note that these features correspond to the means of the nominal features, and that the DP finds the actual transitions without error.

Figure 5(a) shows the nominal scalar LDA features and actual transition points for a radio broadcast stream consisting of commercials, a full song, DJ talk, and the start of another song. These features show some minor trends but no definitive content discrimination; this is because the stream content is substantially different from the training set and indeed includes some speech-on-music regions. Despite the training mismatch, the DP is still able to locate the content transitions. The one apparent error occurs at a point in the song where the dynamics change considerably.

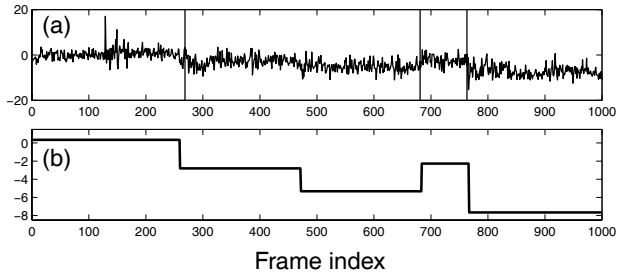

Fig. 5. Plot of (a) the scalar LDA feature and (b) the cluster features derived by the dynamic program for a radio broadcast stream. The solid lines indicate the actual transitions.

Note that the training class densities in Fig. 3(a) suggest a speech/music decision threshold near zero. Such a threshold would accurately classify the commercial as speech but not the speech-on-music DJ talk in the fourth identified segment. From this we can conclude that a more representative training set would be valuable for the strict SMD task, but the identification of speech/music segmentation boundaries can be carried out by the DP robustly even with significantly mismatched prior training.

5. SUMMARY

We have described a two-stage audio segmentation scheme wherein signal features are first extracted and transformed via LDA to optimize cluster scatter and then clustered using DP. The LDA-DP routine converts a feature-space trajectory into a cluster-space trajectory wherein cluster transitions indicate points of significant global change in the signal. The system is general and can be tailored for various applications by appropriate selection of the signal feature set, training set, and DP cost functions. The LDA-DP was shown to provide meaningful content transitions for the speech/music discrimination task even with mismatched LDA training.

6. REFERENCES

[1] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," *Proceedings of IEEE-ICASSP*, vol. 2, pp. 1331–1334, April 1997.

[2] G. Tzanetakis and P. Cook, "Multifeature audio segmentation for browsing and annotation," *IEEE-WASPAA*, Oct. 1999.

[3] J. Foote, "Automatic audio segmentation using a measure of audio novelty," *IEEE-ICME*, July 2000.

[4] M. Goodwin and J. Laroche, "Audio segmentation by feature-space clustering using linear discriminant analysis and dynamic programming," *IEEE-WASPAA*, Oct. 2003.

[5] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search and retrieval of audio," *IEEE Multimedia*, vol. 3, no. 2, pp. 27–36, Fall 1996.

[6] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, San Diego, CA, 1990.

[7] M. Goodwin, "Demo of audio segmentation by clustering," www.atc.creative.com/users/mgoodwin/segment.html.

[8] S. Chen and P. Gopalakrishnan, "Speaker, environment, and channel change detection and clustering via the Bayesian information criterion," *DARPA Broadcast News Wkshp*, 1998.