UNSUPERVISED ANCHOR SPACE GENERATION FOR SIMILARITY MEASUREMENT OF GENERAL AUDIO

Lie Lu[1] *and Alan Hanjalic*[2]

[1]Microsoft Research Asia, Beijing, P.R. China
[2]Department of Mediamatics, Delft University of Technology, Delft, The Netherlands

ABSTRACT

Reliably measuring similarity between audio clips is critical to many applications. As opposed to the conventional way of measuring audio similarity using low-level features directly, in this paper we consider the similarity computation using an *anchor space*. Each dimension of such a space corresponds to a semantic category (anchor). Mapping an audio clip onto this space results in a vector, which indicates the membership probability of this audio clip with respect to each semantic category. The more similar the mappings of two audio clips, the more similar they are. While an anchor space is typically generated in a supervised fashion, supervised approach is infeasible in many realistic scenarios where audio content semantics is too diverse or simply unknown *a priori*. We therefore propose an unsupervised approach to anchor space generation. There, spectral clustering is employed to cluster the audio clips with similar low-level features and then the obtained clusters are adopted as semantic categories. Using this semantic space for audio similarity computation shows a considerable accuracy improvement (7% on mAP) in an audio retrieval system, compared with the conventional low-level feature based approach.

Index Terms— *audio similarity computation, anchor space, audio content analysis, audio segmentation*

1. INTRODUCTION

A reliable method for computing similarity between audio clips is the basis for virtually all higher-level processing steps and applications involving audio signals, such as content-based audio retrieval, categorization, recommendation, and personalization. An audio clip can be as short as an *audio segment* (several seconds), or as long as an *audio document* (several minutes). As can be seen from previous works on audio classification [9], audio effects detection [3], and audio event and scene analysis [4][12], audio content similarity is typically computed by directly comparing audio clips in terms of low-level features. However, while this may work well for short audio segments (e.g. as typically the case in classifying audio segments into speech, music and noise), this is not likely to work well in the case of longer audio documents due to the richness of signal mixtures and their strong variations over time. Clearly, a more sophisticated representation scheme needs to be found for long audio documents, which can neglect irrelevant signal variations and reveal their high-level similarity.

As an alternative to the conventional approaches to audio similarity computation, an *anchor space* can be created, the dimensions of which correspond to various (mid-level) *semantic categories*. Typical examples of such categories are *audio elements* [2][8] or *audio keywords* [12], which we created and used in our previous work to compute the similarity of short audio segments for the purpose of composite audio analysis and auditory scene clustering. As an analogy to this, a longer audio document could also be represented in a suitable anchor space, just as a text document can be represented by a vector of words and their weights. Such an approach therefore enables a more reliable analysis of long audio documents, due to suppression of irrelevant temporal fluctuations of audio signal composition. If an audio document is mapped onto the anchor space, the value obtained for each dimension indicates the membership probability of this audio document with respect to the corresponding semantic category. The more similar the membership of two audio documents in each semantic category, the more similar these two audio documents are.

An anchor space can be created in a supervised or an unsupervised fashion. A supervised approach usually reaches high accuracy and allows control of the semantic level at which anchors are defined [1]. As shown by Berenzweig et al. [1], the selection of anchor in the case of music classification and similarity computation can be done even at a level as high as artist names and music genres. However, the supervised approach is infeasible if processing an unknown composite audio document, or if audio content semantics is too complex (diverse) to easily select appropriate anchors. With the objective of expanding the applicability of the anchor space concept onto a general audio content analysis case, and based on our previous work [2][8] on unsupervised content analysis in general composite audio signals, we propose in this paper an unsupervised method for building an anchor space. While our previous work focused on auditory scenes clustering within one audio document, in this paper we consider the computation of similarity between longer audio documents.

We start the technical part of this paper in Section 2, by explaining the low-level feature extraction step, which forms the basis of anchor space generation. Then, Section 3 reviews the conventional methods for feature-based audio similarity computation. In Section 4, we focus on the challenging problem of building an anchor space for a general composite audio in an unsupervised fashion. In Section 5 we evaluate the proposed approach, by comparing it to the conventional (feature-based) methods and with an anchor space created in a supervised fashion. Section 6 concludes the paper.

2. AUDIO FEATURE EXTRACTION

To extract low-level features, an audio document is first divided into frames of 25ms with 50% overlap. Inspired by previous works on content-based audio classification and audio effect de-

tection [2][9], a set of features is extracted to characterize an audio frame, including short-time energy, zero-crossing rate, sub-band energy ratios, brightness, bandwidth, pitch, pitch periodicity, 8-order Mel-frequency cepstral coefficients (MFCCs), sub-band spectral flatness, sub-band spectral flux and harmonicity prominence. In our approach, the spectral domain is equally divided into 8 sub-bands in Mel-scale and then the sub-band features are extracted. All the above features are collected into a 39-dimensional feature vector per audio frame.

In order to reduce the computational complexity, we choose to group audio frames into temporal audio segments with a sliding window of 1 second with 0.5 seconds overlap, and to use these audio segments as the basis for the subsequent steps. At each window position, the mean and standard deviation of the frame-based features are computed and used to represent the corresponding one-second-long audio segment (78 dimensions).

Since the characteristics and dynamics of each feature component are quite different, a normalization process is performed on each feature component to make their value ranges similar. The normalization is processed as $x_i^{'} = (x_i - \mu_i)/\sigma_i$, where x_i is the i-th feature component, and where the corresponding mean μ_i and standard deviation σ_i can be calculated from a development dataset. Besides normalization, we also employ principle component analysis (PCA) to further reduce the dimensionality of the relevant feature space. In our approach, a reduction from the original 78 dimensions to 45 dimensions was reached.

3. FEATURE-BASED AUDIO CLIP SIMILARITY

Table 1 summarizes some typical approaches for computing audio similarity using low-level features directly. Depending on how the features are modeled and used for similarity computation, we distinguish here three basic classes of approaches, which we label as *Mean*, *Gaussian* and *GMM*:

- *Mean*: an audio clip is represented by averaging the feature vectors. Various distance metrics, such as L1, can be used to measure the similarity in this case.
- *Gaussian*: an audio clip is represented by a Gaussian feature model. Here, KL divergence (KLD) [7] can be used to measure the distance of such models. For symmetry, the distance $D(f,g) = KL(f \parallel g) + KL(g \parallel f)$ is typically used.
- *GMM*: an audio document is represented by a Gaussian Mixture Model (GMM). *KLD* can also be used to measure the similarity between two GMMs. For this purpose, two approximation algorithms can be used, the *pairing* scheme and *unscented transform* [6]:
  - *Pairing*: each Gaussian kernel f_i in GMM f is first paired with the kernel g_j in GMM g which is the most similar to f_i, and then the *KLD* between two GMMs is approximated by the weighted sum of the *KLD* between each kernel pair, as

$$KL(f \parallel g) \approx \sum_{i=1}^{n} \alpha_i \min_{j=1}^{m} KL(f_i \parallel g_j) \qquad (1)$$

  - *Unscented transform:* unscented transform is usually used to obtain a better alternative to the extended Kalman filter. Following [6], the *KLD* between two *GMMs* is approximated as,

$$KL(f \parallel g) \approx \frac{1}{2D} \sum_{i=1}^{n} \alpha_i \sum_{k=1}^{2D} \log \frac{p(x_{i;k} \mid f)}{p(x_{i;k} \mid g)} \qquad (2)$$

where $x_{i;k}$ is the kth "sigma" point in the i-th kernel $f_i = \mathcal{N}(\mu_i, \Sigma_i)$, i.e., $x_{i;k} = \mu_i + \sqrt{D}\sigma_{i;k}$ and $x_{i;k+D} = \mu_i - \sqrt{D}\sigma_{i;k}$ ($1 \le k \le D$), and D is the feature dimension; $\Sigma_i = (\sigma_{i;1}^2, \sigma_{i;2}^2, ..., \sigma_{i;D}^2)$ is the diagonal covariance matrix; and $p(x_{i;k} \mid f)$ is the probability density function of a GMM.

- *GMM with tied covariance*. Some audio documents may not have *enough* segments to accurately estimate the covariance parameters of each Gaussian kernel. *Parameter tying*, the technique often used in speech recognition, can be employed in this case. Covariance parameters are tied and trained together by all the segments of an audio document.

Table 1: Low-level feature based audio document representation and corresponding similarity measure

Document Representation	Similarity Measure
Mean	*L1*
Gaussian	$KL(f \parallel g) + KL(g \parallel f)$
GMM (w/o parameter tying)	*KLD*, unscented transform
	KLD, pairing based algorithm.

4. AUDIO SIMILARITY BASED ON ANCHOR SPACE

The first step in building an anchor space is to select the set of anchors (i.e., space dimensions). If we denote the n-dimensional anchor set by (C_1, C_2, …, C_n), the mapping of an audio document onto this anchor space can be represented by the vector

$$\left[p(C_1 \mid d), \quad p(C_2 \mid d), \quad ..., \quad p(C_n \mid d)\right] \qquad (3)$$

Here, $p(C_i \mid d)$ represents the membership (posterior probability) of the audio document d with respect to the anchor (semantic class) C_i. The probability $p(C_i \mid d)$ can be further calculated as following, supposing the prior $p(C_i)$ is uniformly distributed,

$$p(C_i \mid d) = p(d \mid C_i)\, p(C_i) / p(d) = \frac{p(d \mid C_i)}{\sum_i p(d \mid C_i)} \qquad (4)$$

and the likelihood $p(d \mid C_i)$ is calculated as,

$$p(d \mid C_i) = p(s_1, ..., s_N \mid C_i) = \prod_{k=1}^{N} p(s_k \mid C_i) \qquad (5)$$

where s_k is the k-th audio segment in the audio document d; N is the segment number; $p(s_k \mid C_i)$ is the segment likelihood given the semantic category C_i.

Compared with the above document-level "normalization" (i.e., normalize on $p(d \mid C_i)$ as (4)), we can also employ segment-level normalization, that is, mapping each audio segment onto the anchorspace, then normalizing on the likelihood vector $<p(s_k \mid C_i)>$ for each audio segment, and finally obtaining the audio document representation as the mean of the audio segment representation,

$$p(C_i \mid d) \propto \frac{1}{N} \sum_{k=1}^{N} p(C_i \mid s_k) \qquad (6)$$

where $p(C_i \mid s_k) = \frac{p(s_k \mid C_i)}{\sum_i p(s_k \mid C_i)}$ is the posterior probability of each audio segment given the semantic category C_i. In this case, the idea is to first assign each audio segment to semantic classes, and then the document representation indicates how many segments are assigned to each semantic class.

Independent of which approach is employed, the representation vector (3) makes it possible to employ *KLD* for computing the distance between two audio documents. The more similar the mappings of two documents, the more similar they are.

4.1 Supervised Anchor Space Generation

Having available a set of pre-defined semantic classes and sufficient manually labeled training data, a number of supervised learning techniques can be used to train the semantic class C_i. These techniques include support vector machines (SVM), hidden Markov model (HMM), GMM, and neural networks [1]. In this paper, each semantic class is modeled by a GMM, so that the probability $p(s_k|C_i)$ can easily be computed.

4.2 Unsupervised Anchor Space Generation

Supervised approaches usually reach high accuracy in modeling and classification of data. However, in many realistic scenarios, no information about the content carried by the processed data is available. In these cases, no (complete) set of semantic classes representative for the dataset can be predefined, and no appropriate training data can be collected. In order to expand the applicability of the anchor space concept onto a general composite audio, we propose an unsupervised approach to automatically discover and build the anchors.

In general, clustering techniques can be applied, and the audio segments with similar low-level features can be clustered together and adopted as a semantic class. However, traditional clustering algorithms, such as *K*-means, are based on the assumption that the cluster distributions in the feature space are Gaussians [5], which is usually not satisfied in complex cases. Also, the cluster results are usually affected by the pre-selected centroids, so that multiple restarts are needed to obtain the optimal results. As a promising alternative, spectral clustering [10] showed its effectiveness in a variety of complex applications, such as image segmentation [13] and the multimedia signal clustering [11]. Following [2], we also choose to employ spectral clustering in our approach, and adopt the self-tuning strategy [14] to further improve the robustness of the clustering process.

In the spectral clustering algorithm [10], an affine matrix is first derived from the data corps, which measures the similarity between each pair of data points (in our case, one-second-long audio segments); then a SVD is performed to extract the eigenvectors and map the original data into a low-dimensional space that can be easily clustered. Now, a practical issue to be resolved is the size of the affine matrix. If there are 300 audio clips in the development set, and if each clip has 3 minutes (i.e. around 360 audio segments), the size of the affine matrix will be around $(300*360)^2$ *4B > 40GB (each value in the matrix is a 4-byte float). Such a matrix is impractical to handle and slows down the SVD considerably. To resolve this, a simplification scheme is proposed: Instead of using all feature vectors, we represent each audio document by the mean vector only (averaging the feature vectors therein), and then apply spectral clustering on the set of the mean vectors computed for the entire data set. The obtained clusters are then adopted as anchors.

Regarding the number of clusters to be formed, spectral clustering proposes an estimation approach based on the eigen-gap [10]. However, in our approach, we manually set various cluster numbers to investigate its effect in the final similarity measure.

5. EVALUATIONS

For our experiments, we collected around 3000 audio documents that were extracted as sound tracks of the video clips from MSN Video. Each audio document usually lasts 2-5 minutes, and is associated with a category (also obtained from MSN Video). There are in total 15 categories, including *Autos*, *Business*, *Entertainment*, *Games*, *Live Music*, *Sports*, *Weathers*, and so on. We randomly chose 300 documents as a development set for anchor space building. The rest of the audio documents are kept as the test set.

We test our anchor space building method on a retrieval scenario, for which an audio retrieval system is built. As evaluation strategy, we apply a leave-one-out-like approach, that is, we select each audio document in the test set as a query, after which all other audio documents are ranked based on the proposed similarity measure. The documents belonging to the same category are assumed similar in our experiments. *Mean average precision* (*mAP*), a common metric in information retrieval, is employed to quantify the retrieval performance. The *mAP* is actually the mean value of the average precisions (*AP*) computed for each query separately. To obtain the *AP* value for a particular query, the precision is first computed at each relevant document retrieved, and then these precisions are averaged over the entire test data set. Clearly, the more relevant documents occur higher in the ranked document list, the higher the *AP*. The *AP* value per query can be computed using the expression,

$$AP = \sum_{r=1}^{M} P(r) \times rel(r) \tag{7}$$

where r is the rank, M is the size of the test set, $rel(r)$ is a binary function indicating the relevance of the audio document at rank r with respect to the query, and $P(r)$ is the precision at top r returned documents.

Next to the *mAP*, the *mAP@N* is also evaluated, which represents the mean average precision at the top N ranks (similar to (7), but with a fixed N replacing M). The latter metric may be practically useful since the users are usually ready to review only the first N retrieved documents and do not want to check the entire data set.

Table 2: Evaluation of low-level feature based audio document similarity measure (%), where Mean, GAU and GMM are three representation schemes; tiecov: using parameter tying for covariance parameter estimation; pair: paring scheme, ut: unscented transform.

	mAP	*mAP25*	*mAP50*	*mAP100*
[Mean, L1]	41.4	71.9	66.8	61.2
[GAU, KL]	43.3	72.6	67.6	62.1
[GMM, KL.pair]	39.5	69.6	64.2	58.5
[GMM, KL.ut]	40.7	70.1	64.9	59.3
[GMM.tiecov, KL.pair]	43.7	72.9	68.0	62.5
[GMM.tiecov, KL.ut]	43.9	73.2	68.2	62.7

Table 2 shows the retrieval performance when using the similarity measures based on low-level features (in Section 3). As expected, the Gaussian feature models work better than a simple averaging of the feature vectors (2% improvement on *mAP*). However, the *mAP* decreases when audio documents are modeled

with GMM. This is mainly due to the insufficient data available to model the covariance at each kernel. Using the parameter tying scheme, the *mAP* improves for 3%, compared with the untying one. The table also shows the comparison between the unscented transform and the pairing scheme to realize the approximate *KLD* between two GMMs. The *mAP* obtained by these two approximate approaches is very similar in our cases.

Table 3: Evaluation of the anchor space based audio document similarity measure (%), comparing the supervised approach (sup) vs. unsupervised anchor space building (unsup); segment-level normalization (segl) vs. document-level normalization (docl); and various cluster numbers (as the first number in the first column indicates)

	mAP	mAP25	mAP50	mAP100
[15. sup, docl]	61.3	73.2	71.4	69.3
[15. sup, segl]	58.7	77.3	74.3	71.2
[10, unsup, docl]	44.0	62.1	58.3	54.7
[10, unsup, segl]	45.0	65.6	61.7	57.8
[16, unsup, docl]	46.0	65.4	61.5	57.7
[16, unsup, segl]	48.5	70.6	66.7	62.7
[20, unsup, segl]	49.7	71.4	67.6	63.9
[24, unsup, segl]	50.7	72.5	68.8	65.1
[28, unsup, segl]	50.4	73.0	69.2	65.2

Table 3 shows the retrieval performance when using the similarity measure based on an anchor space. We tested this for both supervised and unsupervised case. In the implementation of the supervised approach, we selected 15 anchors, where each anchor corresponds to each category, and is modeled by a GMM with 4 mixtures. With this setup, the performance has dramatically improved compared to the low-level feature based approach. The *mAP* is up to 61.3% and improves for 18%.

In evaluating our unsupervised anchor space building approach, we tried different cluster numbers, including 10, 16, 20, 24, and 28; and each clustered semantic class is also modeled as a GMM with 4 mixtures The best *mAP* (50.7%) is achieved with the cluster number 24 and with segment-level normalization. It achieves 7% absolute improvement compared with the low-level features based approach, which shows the effectiveness of the unsupervised approach. However, compared with the supervised case, the *mAP* still has 10% lower retrieval performance. This indicates that there is still much room to improve the unsupervised approach. We need to discover which additional information could be potentially helpful and then employ it in anchor space building.

Regarding the comparison between the segment-level normalization and document-level normalization, the former performs slightly better in the unsupervised case, while the latter is better in the supervised case. This can be explained in the following way. In the supervised approach, the anchor space is built on the categories, and a category is also the ground-truth to measure if two audio documents are similar. From this point of view, the supervised learning approach presented in this paper is more like audio document classification. Therefore, to calculate the posterior $p(C_i \mid d)$, that is to perform document-level normalization, works better in this case. However, in the unsupervised anchor space building, each semantic class is similar to an audio element. Thus, an audio document is better represented by a vector indicating the ratio of audio segments assigned to each semantic class. The posterior $p(C_i \mid s)$, that is, segment-level normalization, is a necessary step for such a document representation.

6. CONCLUSION

This paper proposes a novel approach to unsupervised anchor space generation for the purpose of audio document similarity measurement. We employ spectral clustering to group audio segments with similar low-level features, after which the obtained clusters are adopted as anchors. Mapping an audio document onto such an anchor space and then computing the similarity between two audio documents based on their mappings leads to an improvement of up to 7% *mAP* in an audio retrieval scenario, compared to the conventional feature-based similarity measures. Our comparison of this unsupervised approach with supervised anchor space generation showed there is still some room to improve the unsupervised approach. For, example, we may use GMM to represent each audio clip and measure their similarity for spectral clustering. It might generate better anchor models, so as to improve the mAP of audio retrieval.

REFERENCES

[1] Berenzweig A, Ellis D. "Anchorspace for classification and similarity measurement of music", *Proc. ICME* 2003, 29-32

[2] Cai R, Lu L, Hanjalic A. "Unsupervised Content Discovery in Composite Audio", *Proc. ACM Multimedia 05*, 628-637, 2005

[3] Cai R, Lu L, Hanjalic A, Zhang H-J, and Cai L-H. "A flexible framework for key audio effects detection and auditory context inference", *IEEE Trans. Audio, Speech and Language Processing,* 14 (3), 1026-1039, 2006

[4] Cheng W-H, Chu W-T, and Wu J-L. "Semantic context detection based on hierarchical audio models". *Proc. ACM SIGMM Workshop on Multimedia Information Retrieval* 2003, 109-115.

[5] Duda RO, Hart PE, and Stork DG. *Pattern Classification, Second Edition*. John Wiley & Sons, NJ, 2000.

[6] Goldberger J, and Aronowitz H. "A Distance Measure between GMMs based on the Unscented Transform and its Application to Speaker Recognition", *Proc. InterSpeech 2005*

[7] Kullback S., *Information Theory and Statistics*, Dover Publications, New York, 1968.

[8] Lu L, Hanjalic A. "Towards Optimal Audio Keywords Detection for Audio Content Analysis and Discovery", *Proc. ACM Multimedia 06*, 825-834, 2006

[9] Lu L, Zhang H-J, and Jiang H. "Content analysis for audio classification and segmentation". *IEEE Trans. Speech Audio Processing*, 10 (7), 504-516, 2002

[10] Ng AY, Jordan MI, and Weiss Y. "On spectral clustering: analysis and an algorithm". *Advances in Neural Information Processing Systems* (*NIPS) 14*, 2001, 849-856.

[11] Ngo C-W, Ma Y-F, and Zhang H-J. "Video summarization and scene detection by graph modeling". *IEEE Trans. Circuits and Systems for Video Technology*, 15 (2), 296-305, 2005.

[12] Xu M, Maddage N, Xu CS, Kankanhalli M, and Tian Q. "Creating audio keywords for event detection in soccer video". *Proc. ICME* 2003, Vol. 2, 281-284.

[13] Yu SX, and Shi J. "Multiclass spectral clustering". *Proc. ICCV* 2003, Vol. 1, 313-319.

[14] Zelnik-Manor L, and Perona P. "Self-tuning spectral clustering". *NIPS*17, 2004, 1601-1608.