AUTO-TAGGING MUSIC CONTENT WITH SEMANTIC MULTINOMIALS

Luke Barrington, Douglas Turnbull*, Gert Lanckriet
Dept. of Electrical and Computer Engineering *Dept. of Computer Science and Engineering
Computer Audition Laboratory
University of California, San Diego
lukeinusa@gmail.com, dturnbul@cs.ucsd.edu, gert@ece.ucsd.edu

ABSTRACT

We present a system for automatically associating music content with relevant semantic tags. Our supervised multi-label model (SML) consists of one Gaussian mixture model (GMM) distribution over an audio feature space for each tag in our vocabulary. Using the SML model, we annotate a novel song with a *semantic multinomial*: a normalized vector of likelihoods for a song's audio features under each of these tag GMMs. When a tag GMM assigns high probability to the audio features, it is likely that the tag is relevant to that song. Tag GMMs are learned in an efficient manner using the mixture hierarchies expectation-maximization algorithm which combines song GMMs (learned from individual training songs) for all songs that have been associated with the tag.

The system describe in this extended abstract showed the top overall performance in the 2008 Music Information Retrieval Evaluation eXchange (MIREX) "Audio Tag Classification" task. A more detailed description of our approach can be found in:

D. Turnbull, L. Barrington, D. Torres, G. Lanckriet **Semantic Annotation and Retrieval of Music and Sound Effects** *IEEE Transactions on Audio, Speech, and Language (TASLP)*, pp. 467-476, February 2008.

1 MODELING AUDIO AND SEMANTICS

Our auto-tagging music information retrieval (MIR) system takes an audio track as a query and ranks all tags in the vocabulary by relevance to the track. For example, given the song "Hey Jude" by the Beatles, our system outputs:
"This is a **pleasant pop** song that also has a **rock** feel. It features **acoustic guitar**, **piano** and **synthesizer**. The vocals are **high-pitched** and **emotional**. It is a song with **slow tempo** and **positive feelings** that you might like to listen to while **getting ready to go out**."
where words in bold are the relevant tags chosen by the auto-tagging system.

The system is based on the models of [5, 2] which have been applied to the domains of audio and image retrieval respectively. Audio models are learned from a database of audio tracks with associated text captions that describe the audio content:

$$\mathcal{D} = \{(\mathcal{A}^{(1)}, \mathbf{c}^{(1)}), ..., (\mathcal{A}^{(|\mathcal{D}|)}, \mathbf{c}^{(|\mathcal{D}|)})\} \quad (1)$$

where $\mathcal{A}^{(s)}$ and $\mathbf{c}^{(s)}$ represent the s-th song and the associated text caption, respectively. Each tag comes from a fixed vocabulary, $\mathcal{V}$.

1.1 Modeling Audio Tracks

The audio data for a single song (or song clip) is represented as a *bag-of-feature-vectors*, i.e., an unordered set of feature vectors $\mathcal{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_{|\mathcal{A}|}\}$ that are extracted from the audio signal. For each 22050Hz-sampled, monaural audio track, we compute the first 13 Mel-frequency cepstral coefficients as well as their first and second instantaneous derivatives for each half-overlapping short-time ($\sim$12 msec) segment [1], resulting in about 5000 39-dimensional feature vectors per 30 seconds of audio content.

Each song, s, is compactly represented as a probability distribution over the audio feature space, $P(\mathbf{a}|s)$. The song distribution is approximated as a K-component Gaussian mixture model (GMM);

$$P(\mathbf{a}|d) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{a}|\mu_k, \Sigma_k),$$

where $\mathcal{N}(\cdot|\mu, \Sigma)$ is a multivariate Gaussian distribution with mean μ and covariance matrix Σ, and π_k is the weight of component k in the mixture. In this work, we consider only diagonal covariance matrices since using full covariance matrices can cause models to overfit the training data, while scalar covariances do not provide adequate generalization. The parameters of the GMM are learned using the Expectation Maximization (EM) algorithm [3].

1.2 Modeling Semantic Tags

The set of tags or caption for a track, $\mathbf{c}$, is a *bag of words*, represented as a binary vector, where $\mathbf{c}_i = 1$ indicates that tag t_i is associated with this song. While various methods

have been proposed for annotation of music [5, 7] and animal sound effects [4], we follow the work of [5, 2] and learn a GMM distribution for each tag t_i in the vocabulary. In particular, the distribution of audio features for tag t_i is an R-component GMM;

$$P(\mathbf{a}|t_i) = \sum_{r=1}^{R} \pi_r \mathcal{N}(\mathbf{a}|\mu_r, \Sigma_r), \quad (2)$$

The parameters of the tag-level distribution, $P(\mathbf{a}|t_i)$, are learned using the audio features from every track s, that has t_i in its caption $\mathbf{c}^{(s)}$. That is, the training set $\mathcal{T}_i$ for tag t_i consists of only the *positive* examples:

$$\mathcal{T}_i \quad = \quad \{\mathcal{A}^{(s)} \; : \; \mathbf{c}_i^{(s)} = 1, s = 1, \ldots, |\mathcal{D}|\}$$

Learning the tag distribution directly from all the feature vectors in $\mathcal{T}_i$ can be computationally intensive. Hence, we adopt the strategy of [2] and use an extension of EM, the hierarchical EM algorithm [6], to efficiently and robustly learn tag-level distributions $P(\mathbf{a}|t_i)$ from all the song-level distributions $P(\mathbf{a}|s)$ associated with tag t_i.

The final semantic model is a collection of tag-level distributions $P(\mathbf{a}|t_i)$, that models the distribution of audio features associated with the semantic concept t_i.

2 AUTO-TAGGING

Given a set of tag models, the auto-tagging system works by extracting features for a new song clip and evaluating their likelihood under each of the tag models as in Equation 2. To combine the likelihoods of each feature into an estimate of the clip likelihood, we assume that the features are conditionally independent, given the tag and so:

$$P(\mathcal{A}|t_i) = \prod_{j=1}^{|\mathcal{A}|} P(\mathbf{a}_j|t_i).$$

We use Bayes rule to convert this song likelihood into a posterior tag probability:

$$P(t_i|\mathcal{A}) = \frac{P(\mathcal{A}|t_i)P(t_i)}{\sum_{j=1}^{|\mathcal{V}|} P(\mathcal{A}|t_j)P(t_j)}. \quad (3)$$

Computing the posterior probabilities of each tag from a fixed vocabulary $\mathcal{V}$ allows us to represent an audio track as a *semantic feature vector*, where each feature represents the relevance of each tag. For example, the semantic representation of the song "Heartbreak Hotel" by Elvis Presley might have high values in the "blues", "guitar" and "mournful" semantic dimensions, and low values for "electronica", "clarinet" and "jolly".

The semantic feature vector is computed using an annotation system that assigns a weight to each semantic concept. Although any annotation system that outputs weighted labels could be used, when using the probabilistic tag models described in the previous section, the semantic feature vectors are multinomial distributions with each feature equal to the posterior probability of that tag occurring, given the audio features. Formally, given the audio features $\mathcal{A}$, the semantic multinomial is $\pi = \{\pi_1, \ldots, \pi_{|\mathcal{V}|}\}$ with each entry given by:

$$\pi_i = P(t_i|\mathcal{A}),$$

as defined in Equation 3. All the tags in the vocabulary can now be ordered by their relevance to a given song by sorting the song's semantic multinomial. The most relevant tags for a given song are now found by picking the peaks of this semantic multinomial that lie above a given threshold (e.g., calculated based on the tag's prior).

3 MIREX AUDIO TAG CLASSIFICATION

For the MIREX 2008 Audio Tag Classification competition, the UCSD Computer Audition Laboratory system has been packaged as a set of MATLAB functions. The first function, *extractFeatures.m*, reads a text file listing audio file names, extracts features from these files and learns song-level GMMs for each file. The second function, *TrainAndClassify.m*, reads a text file that lists which tags are associated with the training song and learns tag-level GMMs for each tag. These GMMs are then used to automatically tag the unlabeled test songs. This results in a probability of associating each tag with each song which is output as an affiinity matrix. A further binary matrix is output, indicating the tags that are relevant to each song. This binary matrix is calculated by thresholding the affinity matrix such that the number of testing songs associated with a given tag is proportional to the frequency with which that tag was applied to the training songs.

3.1 Results

Six teams submitted eleven entries to the auto-tagging contest. There were three metrics that evaluated the **retrieval performance** of the system:

Metric	Score	Ranking
F-measure	0.28	1
Average Tag Accuracy	0.90	2 (1st place = 0.91)
AUC ROC-Tag	0.77	1

One metric evaluated **annotation performance**:

A full description of the results can be found at: `http://www.music-ir.org/mirex/2008/results/MIREX2008_overview_A0.pdf`

Metric	Score	Ranking
AUC ROC-Clip	0.84	1

4 REFERENCES

[1] C. R. Buchanan. Semantic-based audio recognition and retrieval. Master's thesis, School of Informatics, University of Edinburgh, 2005.

[2] G. Carneiro and N. Vasconcelos. Formulating semantic image annotation as a supervised learning problem. *IEEE CVPR*, 2005.

[3] 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 B*, 39:1 – 38, 1977.

[4] M. Slaney. Mixtures of probability experts for audio retrieval and indexing. *IEEE Multimedia and Expo*, 2002.

[5] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, 16(2), 2008.

[6] N. Vasconcelos. Image indexing with mixture hierarchies. *IEEE CVPR*, pages 3–10, 2001.

[7] B. Whitman and D. Ellis. Automatic record reviews. *ISMIR*, 2004.