MIREX 2010 AUDIO TAG CLASSIFICATION VIA A BAG OF SYSTEMS REPRESENTATION

Katherine Ellis
University of California, San Diego
kellis@ucsd.edu

Emanuele Coviello
University of California, San Diego
ecoviell@ucsd.edu

Gert R.G. Lanckriet
University of California, San Diego
gert@ece.ucsd.edu

ABSTRACT

This paper describes an auto-tagging system presented to MIREX 2011 that represents a "Bag of Systems" (BoS) representation of music. Similar to the Bag of Words representation for text documents, the BoS representation uses a dictionary of musical codewords, where each codeword is a generative model that captures timbral and temporal characteristics of music. Songs are represented as a BoS histogram over codewords, and our system uses multiclass logistic regression on these histograms to learn associations between tags and histograms. Given a new song, the system computes a BoS histogram and outputs a vector of tag-affinities, called a semantic multinomial. Compared to estimating a single generative model to directly capture the musical characteristics of songs associated with a tag, the BoS approach offers the flexibility to combine different classes of generative models at various time resolutions through the selection of the BoS codewords.

1. THE BAG OF SYSTEMS REPRESENTATION OF MUSIC

Analogous to the Bag of Words representation of text documents, the BoS approach represents songs with respect to a codebook, in which generative models are used in lieu of words. These generative models compactly characterize typical audio features, musical dynamics or other acoustic patterns in songs.

In our system we use two different classes of generative models as codewords: Gaussians and dynamic textures (DTs).

1.1 BoS Codewords

1.1.1 Gaussian codewords

Gaussian codewords capture typical audio features, parameterized by a mean feature vector μ and covariance Σ. In our system we use 39-dimensional Mel-frequency cepstral coefficients appended with first and second derivatives (MFCC-delta), extracted from half-overlapping 46 ms windows of audio as feature vectors for Gaussian codewords.

1.1.2 Dynamic Texture codewords

DT codewords explicitly model the temporal dynamics of audio by modeling ordered sequences of audio features vectors. A DT treats an audio fragment $y_{1:\tau}$ (i.e., a sequence of τ audio feature vectors) as the output of a linear dynamical system (LDS):

$$x_t = Ax_{t-1} + v_t, \quad (1)$$

$$y_t = Cx_t + w_t + \bar{y}, \quad (2)$$

where the random variable $y_t \in \mathbb{R}^m$ encodes the timbral content (audio feature vector) at time t, and a lower dimensional hidden variable $x_t \in \mathbb{R}^n$ encodes the dynamics of the observations over time. The model is specified by parameters $\Theta = \{A, Q, C, R, \mu, S, \bar{y}\}$, where the state transition matrix $A \in \mathbb{R}^{n \times n}$ encodes the evolution of the hidden state x_t over time, $v_t \sim \mathcal{N}(0, Q)$ is the driving noise process, the observation matrix $C \in \mathbb{R}^{m \times n}$ encodes the basis functions for representing the observations y_n, $\bar{y}$ is the mean of the observation vectors, and $w_t \sim \mathcal{N}(0, R)$ is the observation noise. The initial condition is distributed as $x_1 \sim \mathcal{N}(\mu, S)$.

In our system, feature vectors for DT codewords are 34-bin Mel-frequency spectral features, extracted from half-overlapping 12 ms windows of audio. Since the DT models fragments of audio instead of single feature vectors, we group consecutive feature vectors into fragments $y_{1:\tau}$ consisting of 125 consecutive feature vectors, corresponding to 726 ms of audio. We sample these fragments every 36 ms.

1.2 Codebook generation

To build a codebook, we derive a set of representative codewords from the collection of songs which we call the codebook set $\mathcal{X}_c$. To do this, we learn a mixture model of K_s codewords of each class (Gaussian or DT) from each song in $\mathcal{X}_c$, using the EM algorithm. Each mixture component then becomes a codeword, and we aggregate all these codewords to form the BoS codebook, $\mathcal{V}$, which contains $|\mathcal{V}| = 2K_s|\mathcal{X}_c|$ codewords. We precompute the BoS codebook from songs in the CAL500 [1] dataset in order to save time when running our algorithm.

1.3 Representing songs with the codebook

Once a codebook is available, a song is represented by a codebook multinomial (CBM) $\mathbf{b} \in \mathbb{R}^{|\mathcal{V}|}$ that reports how

often each codeword appears in that song, where $b[i]$ is the weight of codeword i in the song. To build the CBM for a given song, we count the number of occurrences of each codeword in the song by computing its likelihood at various points in the song and comparing it to the likelihood of other codewords derived from the same base model class. To compute the likelihood of a given codeword at a certain point in the song, we extract a fragment of audio information y^t depending on the time scale and model class of the codeword in question. I.e., for Gaussian codewords, y^t is a single audio feature vector, while for DT codewords, y^t is a sequence of 125 feature vectors. We count an occurrence of a codeword if it has one of the k highest likelihoods of all the codewords in that class. Hence we construct the histogram $\mathbf{b}$ for song $\mathcal{Y}$ by counting the frequency with which each codeword $\Theta_i \in \mathcal{V}$ is chosen to represent a fragment:

$$b[i] = \frac{1}{M|\mathcal{Y}_m|} \sum_{y^t \in \mathcal{Y}_m} \frac{1}{k} \mathbb{1}[\Theta_i = \underset{\Theta \in \mathcal{V}_m}{\operatorname{argmax}}{}^k P(y^t|\Theta)] \quad (3)$$

where $\mathcal{V}_m \subseteq \mathcal{V}$ is the subset of codewords derived from the model class m which codeword Θ_i is derived. Normalizing by the number of fragments $|\mathcal{Y}_m|$ (according to class m) in the song, the number of model classes M, and the threshold k leads to a valid multinomial distribution. In our system, we use a threshold of $k = 10$.

2. MUSIC ANNOTATION AND RETRIEVAL USING THE BAG-OF-SYSTEMS REPRESENTATION

Once a BoS codebook $\mathcal{V}$ has been generated and songs are represented by codebook histograms (i.e., CBMs), we annotate songs based on this representation. Given a training set $\mathcal{X}_t$ of CBM-annotation pairs, annotated with semantic tags from a vocabulary $\mathcal{T}$, we use multiclass logistic regression to learn tag-level models. Each song s in $\mathcal{X}_t$ is associated with a CBM $\mathbf{b}_s$ which describes the song's acoustic content with respect to the BoS codebook $\mathcal{V}$. The song s is also associated with an annotation vector $\mathbf{c}_s = (c_1, \ldots, c_{|\mathcal{T}|})$ which express the song's semantic content with respect to $\mathcal{T}$, where $c_i = 1$ if s has been annotated with tag $w_i \in \mathcal{T}$, and $c_i = 0$ otherwise. We train the tag-level model for tag $w_i \in \mathcal{T}$ using all the songs in the training set that have a positive association with that tag. Given the CBM representation of a novel song, $\mathbf{b}$, we can then use the previously trained tag-models to compute how relevant each tag in $\mathcal{T}$ is to the song.

2.1 Multiclass Logistic Regression

Logistic regression defines a linear classifier with a probabilistic interpretation by fitting a logistic function to all CBMs associated to each tag:

$$P(w_i|\mathbf{b}, \beta_i) \propto \exp \beta_i^T \mathbf{b} \quad (4)$$

In our experiments we apply the histogram intersection kernel to the data before learning the tag models, which is defined by the kernel function: $K(\mathbf{a}, \mathbf{b}) = \sum_j min(a_j, b_j)$.

In our implementation we use the software package Liblinear [3] and learn an L_2-regularized logistic regression model for each tag using the "one-vs-the rest" approach. We implement the histogram intersection kernel using software written by Dahua Lin.

For each test song, we collect the posterior probabilities $p(w_i|\mathbf{b})$ and normalize to form a semantic multinomial (SMN) which represents the affinity of each tag for the test song. We choose the ten tags with highest entries in the SMN to have binary relevance for that song.

3. ACKNOWLEDGMENTS

The authors acknowledge support from Qualcomm, Inc., Yahoo! Inc., the Hellman Fellowship Program, NSF Grants CCF-0830535 and IIS-1054960, and the UCSD FWGrid Project, NSF Research Infrastructure Grant Number EIA-0303622.

4. REFERENCES

[1] D. Turnbull, L. Barrington, D. Torres and G. Lanckriet. Semantic Annotation and Retrieval of Music and Sound Effects. *IEEE Transactions on Audio, Speech and Language Processing*. 16(2):467476, 2008.

[2] K. Ellis, E. Coviello, and G. Lanckriet. Semantic Annotation and Retrieval of Music Using a Bag of Systems Representation. *In proc. ISMIR 2011.*

[3] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chi-Jen Lin. Liblinear: A Libarary for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871-1874, 2008.