Identifying Words that are Musically Meaningful

David A. Torres*, Douglas Turnbull, Luke Barrington
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093
{datorres@cs.ucsd.edu, dturnbul@cs.ucsd.edu, lbarrington@ucsd.edu}

Bharath K. Sriperumbudur, Gert R. G. Lanckriet
Department of Electrical and Computer Engineering
University of California, San Diego
La Jolla, CA 92093
{bharathsv@ucsd.edu, gert@ece.ucsd.edu}

Abstract

A musically meaningful vocabulary is one of the keystones in building a computer audition system that can model the semantics of audio content. If a word in the vocabulary is not clearly represented by the underlying acoustic representation, the word can be considered *noisy* and should be removed from the vocabulary. This paper proposes an approach to construct a vocabulary of predictive semantic concepts based on *sparse canonical component analysis* (sparse CCA). Experimental results illustrate that, by identifying these musically meaningful words, we can improve the performance of a previously proposed computer audition system for music annotation and retrieval.

1 Introduction

Over the past two years we have been developing a computer audition system that can annotate songs with semantically meaningful words and retrieve relevant songs based on a text query. This system learns a joint probabilistic model between a vocabulary of words and acoustic feature vectors using a heterogeneous data set of song and song annotations. However, if a specific word is not represented well by the acoustic features, the system will not be able to model this *noisy* word well. In this paper, we explore the problem of *vocabulary selection* for semantic music annotation and retrieval, wherein *acoustic correlation* is used as an indicator for picking candidate words.

Previously, we collected annotations of music using various methods: text-mining song reviews [19], conducting a human survey [20], and exploring the use of a human computation game [21, 23]. In all cases, we are forced to choose a vocabulary using ad-hoc methods. For example, text-mining the song reviews resulted in a list of over 1,000 candidate words which the authors manually pruned if there was a general consensus that a word was not 'musically-relevant'. To collect the survey and game data, we built, a priori, a two-level hierarchical vocabulary by first considering a set of high-level semantic categories ('Instrumentation', 'Emotion', 'Vocal Characteristic', 'Genre') and then listing low-level words ('Electric Guitar', 'Happy', 'Breathy', 'Bebop') for each semantic category. In both cases, a vocabulary required manual construction and included some *noisy* words that degraded the performance of our computer audition system.

A reason that certain words causes problems for our system is that the acoustic representation of these words may be hard to model. This relates to the expressive power of our chosen audio feature

*Corresponding author

representation. For example, if we are interested in words related to long-term music structure (e.g., '12-bar blues') and we only represent the audio using short-term (< 1 sec) audio feature vectors, we may be unable to model such concepts. Another example is words that relate to a geographical association (e.g., 'British Invasion', 'Woodstock') which may have strong cultural roots, but are poorly represented in the audio content.

In our latest research [18], given an audio feature representation, we would like to identify the words that are represented well by the audio content before we try to model them. To do this we propose the use of an unsupervised method based on *canonical correlation analysis* (CCA) to measure *acoustic correlation*. CCA is a method of exploring correlations between two different, but related, vector spaces and has been used in applications dealing with multi-language text analysis [22], learning semantic representations between images and text [4], and localizing pixels which are correlated with audio from a video stream [8]. Similar to the way principal component analysis (PCA) finds informative directions in one feature space by maximizing the variance of projected data, CCA finds directions (projections of the data) across multiple spaces that maximize correlation. Given music data represented in both a semantic feature space and an acoustic feature space, we propose that these directions of high correlation can be used to find words that are strongly characterized by an audio representation.

Specifically, the CCA solution direction, or canonical component, corresponding to the semantic representation of music is a mapping of vocabulary words to weights where a high weight implies that a given word is highly correlated with the audio feature representation. We interpret high weights as singling out words that are "musically meaningful". In other words, the underlying relationship (a correlation) between word and audio feature representation may be more meaningful to model than a word for which no such relationship exists.

Generally CCA returns a mapping of words to weights that is non-sparse, meaning that each word will be mapped to a non-zero weight. Hence, selecting a subset of musically meaningful words from this vocabulary would need to be done by thresholding the weights or some similar heuristic. It is desirable to remove such an arbitrary step from this process. This is done by imposing sparsity on the solution weights, that is we modify the CCA problem so that it tends to give solutions with few non-zero weights (*sparse CCA* [16]). This leads to a very natural criterion for selecting a vocabulary: throw away words with weights equal to zero.

2 Acoustic Correlation with CCA

Canonical Correlation Analysis, or CCA, is a method of exploring dependencies between data which are represented in two different, but related, vector spaces. For example, consider a set of songs where each song is represented by both a *semantic annotation vector* and an *audio feature vector*. An annotation vector for a song is a real-valued (or binary) vector where each element represents the strength of association between the song and a word from our vocabulary. An audio feature vector is a real-valued vector of statistics calculated from the digital audio signal. It is assumed that the two spaces share some joint information which can be captured in the form of correlations between the music data that live in these spaces. CCA finds a one-dimensional projection of the data in each space such that the correlations between the projections is maximized.

More formally, consider two data matrices, $\mathbf{A}$ and $\mathbf{S}$, from two different feature spaces. The rows of $\mathbf{A}$ contain music data represented in the audio feature space $\mathcal{A}$. The corresponding rows of $\mathbf{S}$ contain the music data represented in the semantic annotation space $\mathcal{S}$ (e.g., annotation vectors). CCA seeks to optimize

$$\max_{\mathbf{w}_a \in \mathcal{A}, \mathbf{w}_s \in \mathcal{S}} \{\mathbf{w}_a^T \mathbf{\Sigma}_{as} \mathbf{w}_s \ : \ \mathbf{w}_a^T \mathbf{\Sigma}_{aa} \mathbf{w}_a = 1, \ \mathbf{w}_s^T \mathbf{\Sigma}_{ss} \mathbf{w}_s = 1\} \tag{1}$$

where $\mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}_{aa} & \mathbf{\Sigma}_{as} \\ \mathbf{\Sigma}_{sa} & \mathbf{\Sigma}_{ss} \end{pmatrix}$ is the covariance matrix of $[\mathbf{A}|\mathbf{S}]$.

By formulating the Lagrangian dual of Eq. (1), it can be shown that solving Eq. (1) is equivalent to solving a pair of maximum eigenvalue problems,

$$\mathbf{\Sigma}_{aa}^{-1} \mathbf{\Sigma}_{as} \mathbf{\Sigma}_{ss}^{-1} \mathbf{\Sigma}_{sa} \mathbf{w}_a = \lambda^2 \mathbf{w}_a, \tag{2}$$

$$\mathbf{\Sigma}_{ss}^{-1} \mathbf{\Sigma}_{sa} \mathbf{\Sigma}_{aa}^{-1} \mathbf{\Sigma}_{as} \mathbf{w}_s = \lambda^2 \mathbf{w}_s, \tag{3}$$

with λ being the maximum value of Eq. (1).

The solution vectors, $\mathbf{w}_a$ and $\mathbf{w}_s$, in Eq. (1) are generally non-sparse, meaning that the coefficients in the vectors tend to be non-zero. In many applications it may be of interest to limit the number of non-zero elements in the solution vector as this aids in the interpretability of the results. For example, in this application the solution vector $\mathbf{w}_s$ can be interpreted as a mapping of words to weights where a high weight implies that a given word is highly correlated with the audio feature representation. We impose sparsity on $\mathbf{w}_s$ thereby turning CCA into an explicit vocabulary selection mechanism where words with zero weight (i.e. negligible correlation) are thrown away.

2.1 Sparse CCA

While PCA and its two-view extension, CCA are well studied and understood, to our knowledge, the sparse verion of CCA is not. Different algorithms [7, 26, 2, 16], both convex and non-convex have been proposed for sparse PCA. Recently in [16], we proposed a general framework for solving sparse eigenvalue problems using d.c. programming (difference of convex functions) [5] which we extend here to derive a sparse CCA algorithm.

Consider the variational formulation of CCA in Eq. (1) which can be written as a generalized eigenvalue problem, $\max_{\mathbf{x}} \{\mathbf{x}^T\mathbf{P}\mathbf{x} \ : \ \mathbf{x}^T\mathbf{Q}\mathbf{x} = 1\}$, where $\mathbf{P} = \begin{pmatrix} \mathbf{0} & \mathbf{\Sigma}_{as} \\ \mathbf{\Sigma}_{sa} & \mathbf{0} \end{pmatrix}$, $\mathbf{Q} = \begin{pmatrix} \mathbf{\Sigma}_{aa} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{ss} \end{pmatrix}$ and $\mathbf{x} = \begin{pmatrix} \mathbf{w}_a \\ \mathbf{w}_s \end{pmatrix}$. Therefore, the variational formulation for sparse CCA is given by

$$\max_{\mathbf{x}} \{\mathbf{x}^T\mathbf{P}\mathbf{x} \ : \ \mathbf{x}^T\mathbf{Q}\mathbf{x} = 1, ||\mathbf{x}||_0 \leq k\}, \tag{4}$$

where $1 \leq k \leq n$ and $n = \dim(\mathcal{A}) + \dim(\mathcal{S})$. Eq. (4) is non-convex, NP-hard and therefore intractable (see [16] for detailed discussion). A related problem to Eq. (4) is given by

$$\max_{\mathbf{x}} \{\mathbf{x}^T\mathbf{P}\mathbf{x} - \rho\,||\mathbf{x}||_0 \ : \ \mathbf{x}^T\mathbf{Q}\mathbf{x} \leq 1\}, \tag{5}$$

where $\rho > 0$ is the penalization parameter that controls sparsity. Usually, $||\mathbf{x}||_0$ is replaced by $||\mathbf{x}||_1$ to achieve convexity. However, in our setting, since $\mathbf{P}$ is indefinite, ℓ_1 approximation does not yield any computational advantage. So, we use a better approximation than ℓ_1, given by $\sum_i^n \log|x_i|$, leading to the following program,

$$\max_{\mathbf{x}} \{\mathbf{x}^T\mathbf{P}\mathbf{x} - \rho\sum_{i=1}^{n}\log|x_i| \ : \ \mathbf{x}^T\mathbf{Q}\mathbf{x} \leq 1\}, \tag{6}$$

This approximation has shown superior performance in sparse PCA experiments [16] and SVM feature selection experiments [24] and has nice theoretical guarantees as to its approximation of the zero norm [24, 16]. With $\mathbf{P}$ being indefinite, Eq. (6) can be reduced to a d.c. program as

$$\min_{\mathbf{x}} \left\{ \mu||\mathbf{x}||^2 - \left(\mathbf{x}^T[\mathbf{P} + \mu\mathbf{I}]\mathbf{x} - \rho\sum_{i=1}^{n}\log|x_i| \right) \ : \ \mathbf{x}^T\mathbf{Q}\mathbf{x} \leq 1 \right\}, \tag{7}$$

where $\mu \geq -\lambda_{max}(\mathbf{P})$. Using the d.c. minimization algorithm (DCA) [17], we get the sparse CCA algorithm given by Algorithm 1, which is a sequence of quadratic programs. In our setting, we need to impose a sparsity constraint only on the semantic space, $\mathcal{S}$ and so we need a constraint on $||\mathbf{w}_s||_0$ instead of the overall vector, $||\mathbf{x}||_0$. In such a case, Eq. (8) can be modified as

$$\begin{aligned} \bar{\mathbf{x}}^* = \arg\min_{\bar{\mathbf{x}}} \quad & \mu\bar{\mathbf{x}}^T\mathbf{D}^2(\mathbf{x}_l)\bar{\mathbf{x}} - 2\mathbf{x}_l^T[\mathbf{P} + \mu\mathbf{I}]\mathbf{D}(\mathbf{x}_l)\bar{\mathbf{x}} + ||\boldsymbol{\tau} \circ \bar{\mathbf{x}}||_1 \\ \text{s.t.} \quad & \bar{\mathbf{x}}^T\mathbf{D}(\mathbf{x}_l)\mathbf{Q}\mathbf{D}(\mathbf{x}_l)\bar{\mathbf{x}} \leq 1 \end{aligned} \tag{9}$$

where $\mathbf{D}(\mathbf{x}) = \text{diag}(\mathbf{x})$, $(\mathbf{p} \circ \mathbf{q})_i = a_i b_i$ and $\boldsymbol{\tau} = [0, 0, \overset{\dim(\mathcal{A})}{\cdots}, 0, \rho, \rho, \overset{\dim(\mathcal{S})}{\cdots}, \rho]^T$.

For our purposes Eq. (9) becomes a vocabulary selection mechanism which takes as its input the semantic and audio feature representations for a set of songs, $\mathbf{S}$ and $\mathbf{A}$, and a penalization parameter ρ. The method returns a set of sparse weights $\mathbf{w}_s$ associated with each word in the vocabulary. Words with a weight of zero are removed from our modeling process.

The non-zero elements of the sparse solution vector $\mathbf{w}_s$ can be interpreted as those words which have a high correlation with the audio representation. Thus, in the experiments that follow, setting values of ρ and solving Eq. (9) reduces to a vocabulary selection technique.

Algorithm 1 Sparse CCA Algorithm

Require: Symmetric $\mathbf{P}$, $\mathbf{Q} \succ 0$ and $\rho > 0$

1: Choose $\mathbf{x}_0 \in \{\mathbf{x} : \mathbf{x}^T\mathbf{Q}\mathbf{x} \leq 1\}$ arbitrarily

2: **repeat**

3:

$$\begin{aligned} \bar{\mathbf{x}}^* = \arg\min_{\bar{\mathbf{x}}} \quad & \mu\bar{\mathbf{x}}^T\mathbf{D}^2(\mathbf{x}_l)\bar{\mathbf{x}} - 2\mathbf{x}_l^T[\mathbf{P} + \mu\mathbf{I}]\mathbf{D}(\mathbf{x}_l)\bar{\mathbf{x}} + \rho||\bar{\mathbf{x}}||_1 \\ \text{s.t.} \quad & \bar{\mathbf{x}}^T\mathbf{D}(\mathbf{x}_l)\mathbf{Q}\mathbf{D}(\mathbf{x}_l)\bar{\mathbf{x}} \leq 1 \end{aligned} \tag{8}$$

4: $\mathbf{x}_{l+1} = \mathbf{x}_l \circ \bar{\mathbf{x}}^*$

5: **until** $\mathbf{x}_{l+1} = \mathbf{x}_l$

6: **return** $\mathbf{x}_l$, $\bar{\mathbf{x}}^*$

3 Representing Audio and Semantic Data

In this section we describe the audio and semantic representations, as well as describe the CAL500 [20] and Web2131 [19] annotated music corpora that are used in our experiments. In both cases, the semantic information will be represented using a single annotation vector $\mathbf{s}$ with dimension equal to the size of the vocabulary. The audio content will be represented as multiple feature vectors $\{\mathbf{a}_1, ..., \mathbf{a}_T\}$, where T depends on the length of the song.

The construction of the matrices $\mathbf{A}$ and $\mathbf{S}$ to solve the sparse CCA follows: Each feature vector in the music corpus is associated with the label for its song. For example, for a given song, we duplicate its annotation vector $\mathbf{s}$ for a total of T times so that the song-label pair may be represented as $\{(\mathbf{s}, \mathbf{a}_1), ..., (\mathbf{s}, \mathbf{a}_T)\}$. To construct $\mathbf{A}$ we stack the feature vectors for all songs in the corpus into one matrix. $\mathbf{S}$ is constructed by stacking all the corresponding annotation vectors into one matrix. If each song has approximately 600 feature vectors and we have 500 hundred songs, then both $\mathbf{A}$ and $\mathbf{S}$ will have about 30,000 rows.

3.1 Audio Representation

Each song is represented as a *bag-of-feature-vectors*: we extract an unordered set of feature vectors for every song, by extracting one feature vector for each short-time segment of audio data. Specifically, we compute dynamic Mel-frequency cepstral coefficients (dMFCC) from each half-overlapping, medium-time ($\sim$743 ms) segment of audio content [11].

Mel-frequency cepstral coefficients (MFCC) describe the spectral shape of a short-time audio frame in a concise and perceptually meaningful way and are popular features for speech recognition and music classification (e.g., [13, 9, 15]). We calculate 13 MFCC coefficients for each short-time (23 msec) frame of audio. For each of the 13 MFCCs, we take a discrete Fourier transform (DFT) over the time series of 64 frames, normalize by the DC value (to remove the effect of volume) and summarize the resulting spectrum by integrating across 4 modulation frequency bands: (unnormalized) DC, 1-2Hz, 3-15Hz and 20-43Hz. Thus, we create a 52-dimensional features vector (4 features for each of the 13 MFCCs) for every $3/4$ segment of audio content. For a five minute song, this results in about 800 52-dimensional feature vectors.

We have also explored a number of alternative feature representations, These include auditory filterbank temporal envelope [11], MFCCs (with and without instantaneous derivatives) [20], chroma features [3], and fluctuation patterns [12]. For our experiments we chose a DMFCC representation since it is compact compared with raw MFCC feature representations and shows good performance on the task of semantic music annotation and retrieval compared with these other representations.

3.2 Semantic Representation

The CAL500 is an annotated music corpus of 500 western popular songs by 500 unique artists. Each song has been annotated by a minimum of 3 individuals using a vocabulary of 174 words. We paid 66 undergraduate music students to annotate our music corpus with semantic concepts. We collected a set of semantic labels created specifically for a music annotation task. We considered

Top 3 words by semantic category		
	Agreement	*Acoustic Correlation*
overall	male lead vocals, drum set, female lead vocals	rapping, at a party, hip-hop/rap
emotion	not angry/agressive, not weird, not tender/soft	arousing/awakening, exciting/thrilling, sad
genre	hip-hop/rap, electronica, world	hip-hop/rap, electronica, funk
instrument	male lead vocals, drum set, female lead vocals	drum machine, samples, synthesizer
general	electric texture, not danceable, high energy	heavy beat, very danceable, synthesized texture
usage	driving, at a party, going to sleep	at a party, exercising, getting ready to go out
vocals	rapping, emotional, strong	rapping, strong, altered with effects
Bottom 3 words by semantic category		
	Agreement	*Acoustic Correlation*
overall	at work, with the family, waking up	not weird, not arousing, not angry/agressive
emotion	not powerful/strong, not emotional, weird	not weird, not arousing, not angry/agressive
genre	contemporary blues, roots rock, alternative folk	classic rock, bebop, alternative folk
instrument	trombone, tamborine, organ	female lead vocals, drum set, acoustic guitar
general	changing energy level, minor key tonality, low song quality	constant energy level, changing energy level, not catchy
usage	at work, with the family, waking up	going to sleep, cleaning the house, at work
vocals	falsetto, spoken, monotone	high pitches, falsetto, emotional

Table 1: Top and bottom 3 words by semantic category as calculated by agreement and acoustic correlation.

135 musically-relevant concepts spanning six semantic categories: 29 instruments were annotated as present in the song or not; 22 vocal characteristics were annotated as relevant to the singer or not; 36 genres, a subset of the Codaich genre list [10], were annotated as relevant to the song or not; 18 emotions, found by Skowronek et al. [14] to be both important and easy to identify, were rated on a scale from one to three (e.g., "not happy", "neutral", "happy"); 15 song concepts describing the acoustic qualities of the song, artist and recording (e.g., tempo, energy, sound quality); and 15 usage terms from [6], (e.g., "I would listen to this song while *driving, sleeping, etc.*"). The 135 concepts are converted to the 174-word vocabulary by first mapping bi-polar concepts to multiple word labels ('Energy Level' maps to 'low energy' and 'high energy'). Then we prune all words that are represented in five or fewer songs to remove under-represented words. Lastly, we construct a real-valued 174-dimensional annotation vector by averaging the label frequencies of the individual annotators. Details of the summarization process can be found in [20]. In general, each element in the annotation vector contains a real-valued scalar indicating the strength of association.

The Web2131 is an annotated collection of 2131 songs and accompanying expert song reviews mined from a web-accessible music database[1] [19]. Exactly 363 songs from Web2131 overlap with the CAL500 songs. The vocabulary consists of 317 words that were hand picked from a list of the common words found in the corpus of song reviews. Common stop words are removed and the resulting words are preprocesses with a custom stemming algorithm. We represent a song review as a binary 317-dimensional annotation vector. The element of a vector is 1 if the corresponding word appears in the song review and 0 otherwise.

4 Experiments

4.1 Vocabulary Pruning using Sparse CCA

Sparse CCA can be used to perform vocabulary selection where the goal is to prune noisy words from a large vocabulary. To test this hypothesis we combined the vocabularies from the CAL500 and Web2131 data sets and consider the subset of 363 songs that are found in both data sets.

Based on our own informal user study, we found that the Web2131 annotations are noisy when compared to the CAL500 annotations. We showed subjects 10 words from each data set and asked them which set of words were relevant to a song. The Web2131 annotations were not much better than

[1] AMG All Music Guide www.allmusic.com

vocab.sz.	488	249	203	149	103	50
# CAL500 words	173	118	101	85	65	39
# Web2131 words	315	131	102	64	38	11
%Web2131	**.64**	**.52**	**.50**	**.42**	**.36**	**.22**

Table 2: The fraction of noisy web-mined words in a vocabulary as vocabulary size is reduced: As the size shrinks sparse CCA prunes noisy words and the web-mined words are eliminated over higher quality CAL500 words.

selecting words randomly to from the vocabulary, whereas CAL500 words were mostly considered relevant.

Because Web2131 was found to be noisier than CAL500, we expect sparse CCA to filter out more of the Web2131 vocabulary. Table 2 shows the results of this experiment. In the experiment we select a value for the sparsity parameter ρ and obtain a sparse vocabulary. Then we record how many noisy Web2131 words comprise the resulting vocabulary. The first column in Table 2 reflects the vocabulary with no sparsity constraints. Because the vocabularies are of different sizes, Web2131 initially comprises .64 of the combined vocabulary size. Subsequent columns in the table show the resulting vocabulary sizes as we increase ρ and, consequently, reduce the vocabulary size.

Noisy Web2131 words are being discarded by our vocabulary selection process at a faster rate than the cleaner CAL500 vocabulary suggesting that Web2131 contains more words that are uncorrelated with the audio representation. The different ways that these data sets were collected reinforces this fact. Web2131 was mined from a collection of music reviews; the words used in a music review are not explicitly chosen by the writer because they describe a song. This is the exact opposite condition under which the CAL500 data set was collected, in which human subjects were specifically asked to label songs in a controlled environment.

4.2 Vocabulary Selection for Music Retrieval

In this experiment we apply our vocabulary selection technique to a semantic music annotation and retrieval system. In brief, our system estimates the conditional probabilities of audio feature vectors given words in the vocabulary, $P(song|word)$. These conditional probabilities are modeled as Gausian Mixture Models. With these probability distributions in hand, our system can annotate a novel song with words from its vocabulary, or it can retrieve an ordered list of (novel) songs based based on a keyword query. A full description of this system can be found in [20].

One useful evaluation metric for retrieval is the area under the ROC (AROC) curve. (A ROC curve is a plot of the true positive rate as a function of the false positive rate as we move down a ranked list of songs given a keyword input.) For each word, its AROC ranges between 0.5 for a random ranking and 1.0 for a perfect ranking. Average AROC is used to describe the performance of our entire system and is found by averaging AROC over all words in the vocabulary.

Words that are difficult to model will tend to have low AROC and bring the average AROC of the system down. We propose to use sparse CCA to remove words that are difficult to model. Specifically, we use sparse CCA to generate sparse vocabularies of full size (180) down to 20. (This is done by solving the sparse CCA program as you sweep the sparsity parameter ρ to give you a range of vocabularies of different sizes). Then the conditional probability densities associated with these words, or word models, are trained and the average AROC of the resulting word models is calculated on a test set of 50 songs held out separately from a training set of 450 songs from the CAL500 data set.

Figure 1 shows that the average AROC of our system improves as sparse CCA selects vocabularies of smaller size. Any increase from the left most point of the figure implies that the technique is removing words which would have had a low AROC (thus bringing the average down). To reiterate, it is exactly these kinds of words (with now AROC) that we presume to be difficult to model.

Also shown on 1 are the results of our training based on two alternative vocabulary selection techniques. One is a random baseline, in which vocabulary size is reduced by randomly removing words from the vocabulary. The other is based on a heuristic we used in [18] in which we assigned to words a score based on the notion of human agreement. (The details can be found in [18].) Briefly,

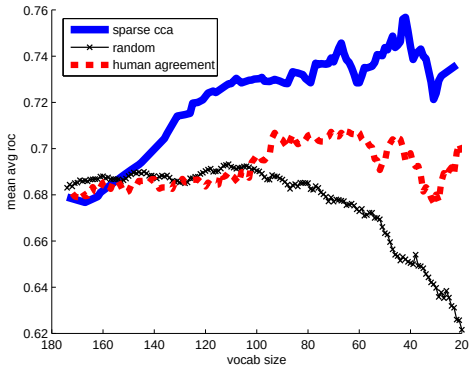

Figure 1: Comparison of vocabulary selection techniques: We compare vocabulary selection using human agreement, acoustic correlation, and a random baseline, as it effects retrieval performance.

because we have access to more than one human annotation per song, we count how many times people agreed with the use of a particular word to describe a song. For example, more subjects tended to agree on more objective instrumentation terms like "this song has a *drum*.", as opposed to more subjective usage terms like "i would listen to this song while *waking up* in the morning." In this case "*drum*" would get a higher *human agreement* score than "*waking up*", and so we posit that it would be easier to model "*drum*" (it would have a higher AROC) than it would to model "*waking up*". Our results show that vocabulary selection using acoustic correlation outperforms using this very logical heuristic of human agreement.

It should be noted that the goal here, in and of itself, is not to raise the performance of some arbitrary system, rather that raising the performance of this system suggests that our vocabulary selection method is removing words that are difficult to model well. Also, as a practical matter, our results show that this technique could be used to guide us as to which words to spend computational resources on.

5 Discussion

We have presented acoustic correlation via sparse CCA as a method by which we can automatically, and in an unsupervised fashion, discover words that are highly correlated with their audio feature representations. Our results suggest that this technique can be used to remove "noisy" words that are difficult to model accurately. This ability to filter poorly represented words also provides a means to construct a musically meaningful vocabulary prior to investing further computational resources in modeling or analyzing the semantics of music as is done in our music annotation and retrieval system. Those interested in the application of vocabulary selection and music analysis are encouraged to view the work of Whitman and Ellis who have previously looked at vocabulary selection by training binary classifiers (e.g., Support Vector Machines) on a heterogeneous data set of web-documents related to artists and the audio content produced by these artists [25].

In future research, we will investigate the impact of using a musically meaningful vocabulary for assessing song similarity through semantic distance. We are interested in developing a query-by-semantic-example system [1] for music which retrieves similar songs by first representing them in the semantic space and then rank-ordering them based on a distance metric in that semantic space. We expect that having a compact semantic representation, which can be found using sparse CCA, we will be able to improve retrieval performance.We also plan to explore the possibility of extracting meaningful semantic concepts from web-based documents through acoustic correlation.

References

[1] Luke Barrington, Antoni Chan, Douglas Turnbull, and Gert Lanckriet. Audio information retrieval using semantic similarity. Technical report, 2007.

[2] A. d'Aspremont, L. El Ghaoui, M.I. Jordan, and G.R.G. Lanckriet. A direct formulation for sparse pca using semidefinite programming. *Accepted for publication in SIAM Review*, 2006.

[3] D. Ellis and G. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. IEEE ICASSP, 2007.

[4] David R. Hardoon, Sandor Szedmak, and Jogn Shawe-Taylor. Canonical correlation analysis; an overview with application to learning methods. *Neural Computation*, 2004.

[5] R. Horst and N. V. Thoai. D.c. programming: Overview. *Journal of Optimization Theory and Applications*, 103:1–43, 1999.

[6] Xiao Hu, J. Stephen Downie, and Andreas F. Ehmann. Exploiting recommended usage metadata: Exploratory analyses. *ISMIR*, 2006.

[7] I. T. Jolliffe, N. T. Trendafilov, and M. Uddin. A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12:531–547, 2003.

[8] Einat Kidron, Yoav Y. Schechner, and Michael Elad. Pixels that sound. In *IEEE Computer Vision and Pattern Recognition*, 2005.

[9] Beth Logan. Mel frequency cepstral coefficients for music modeling. *ISMIR*, 2000.

[10] Cory McKay, Daniel McEnnis, and Ichiro Fujinaga. A large publicly accessible prototype audio database for music research. *ISMIR*, 2006.

[11] M. F. McKinney and J. Breebaart. Features for audio and music classification. *ISMIR*, 2003.

[12] E. Pampalk. *Computational Models of Music Similarity and their Application in Music Information Retrieval*. PhD thesis, Vienna University of Technology, Vienna, Austria, 2006.

[13] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[14] Janto Skowronek, Martin McKinney, and Steven ven de Par. Ground-truth for automatic music mood classification. *ISMIR*, 2006.

[15] M. Slaney. Semantic-audio retrieval. *IEEE ICASSP*, 2002.

[16] Bharath K. Sriperumbudur, David A. Torres, and Gert R. G. Lanckriet. Sparse eigen methods by d.c. programming. In *International Conference on Machine Learning*, 2007.

[17] Pham Dinh Tao and Le Thi Hoai An. D.c. optimization algorithms for solving the trust region subproblem. *SIAM Journal of Optimization*, 8:476–505, 1998.

[18] D. Torres, D. Turnbull, L. Barrington, and G. Lanckriet. Identifying words that are musically meaningful. In *ISMIR 07*, 2007.

[19] D. Turnbull, L. Barrington, and G. Lanckriet. Modelling music and words using a multi-class naïve bayes approach. *ISMIR*, 2006.

[20] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic description using the CAL500 data set. In *To appear in SIGIR '07*, 2007.

[21] D. Turnbull, R. Liu, L. Barrington, D. Torres, and Gert Lanckriet. UCSD CAL technical report: Using games to collect semantic information about music. Technical report, 2007.

[22] Alexei Vinokourov, John Shawe-Taylor, and Nello Cristianini. Inferring a sementic representation of text via cross-language correlation analysis. *Advances in Neural Information Processing Systems*, 2003.

[23] Luis von Ahn. Games with a purpose. *IEEE Computer Magazine*, 39(6):92–94, 2006.

[24] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping. Use of the zero-norm with linear models and kernel methods. *Journal of machine learning research*, 2003.

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

[26] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 2004.