AUTOMATIC RECORD REVIEWS

Brian Whitman
MIT Media Lab
Music Mind and Machine Group

Daniel P.W. Ellis
LabROSA
Columbia University Electrical Engineering

ABSTRACT

Record reviews provide a unique and focused source of linguistic data that can be related to musical recordings, to provide a basis for computational music understanding systems with applications in similarity, recommendation and classification. We analyze a large testbed of music and a corpus of reviews for each work to uncover patterns and develop mechanisms for removing reviewer bias and extraneous non-musical discussion. By building upon work in grounding free text against audio signals we invent an "automatic record review" system that labels new music audio with maximal semantic value for future retrieval tasks. In effect, we grow an unbiased music editor trained from the consensus of the online reviews we have gathered.

Keywords: cultural factors, language, machine learning, audio features, reviews

1. INTRODUCTION

Spread throughout the music review pages of newspapers, magazines and the internet lie the answers to music retrieval's hardest problems of audio understanding: thousands of trained musical experts, known otherwise as reviewers, distill the hundreds of megabytes of audio data from each album into a few kilobytes of semantic classification. Instead of the crude and suspect genre tags and artist names that so often serve as semantic ground truth, we can get detailed descriptions of the audio content (instrumentation, beat, song structure), cultural position (relationships to other groups, buzz, history) and individual preference (the author's opinion of the work). There is tremendous value waiting to be extracted from this data, as the ostensible purpose of a record review is to provide all the necessary categorical and descriptive information for a human judge to 'understand' the recording without hearing it. If we would like to build music intelligences that automatically classify, recommend and even synthesize music for listeners, we could start by analyzing the connection between music (or music-derived audio features) and a listener's reaction as detailed in a review.

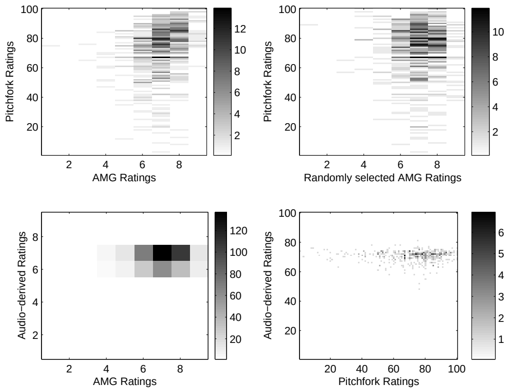
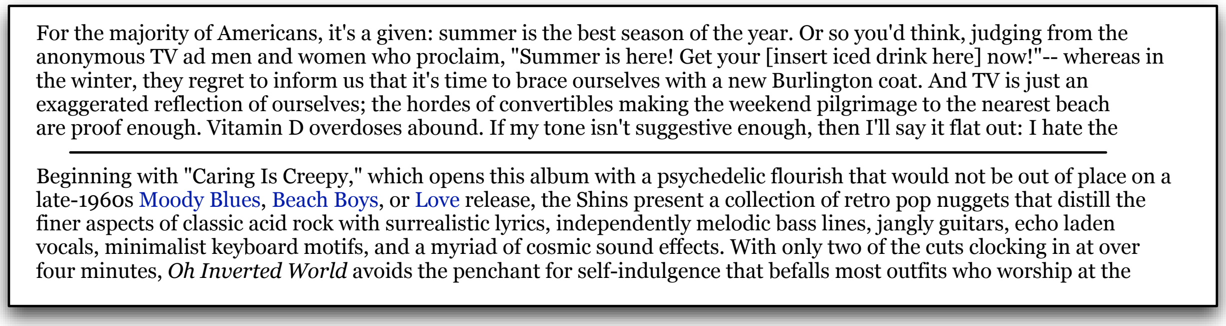


Figure 1. Predicting and analyzing ratings. Top left: correlation of AMG (1-9 scale) to Pitchfork (0-100 scale) ratings, correlation coefficient $r = 0.264$. Top right: Pitchfork ratings to randomly selected AMG ratings, $r = 0.017$ for this instance. Bottom left: predicting AMG ratings from audio features, $r = 0.147$. Bottom right, predicting Pitchfork ratings from audio, $r = 0.127$.

A system for "review understanding" is useful even to text-only retrieval systems: Consider a site that encourages on-line reviews of its stock; user-submitted text can be used in place of a sales-based collaborative filtering recommendation agent, and such systems prove to work well as "buzz" or opinion tracking models[1]. However, in our case we are fortunate to have the subject of the reviews – the music audio itself – simultaneously available, and our work concentrates on the link between description and perception. We believe an audio model of 'romantic interludes' can be far more expressive, informative, and statistically valid than a model of 'Rock' – and given the prospect of scaling our models to hundreds of thousands of terms and phrases applicable to every kind of music, we envision a bias-free computational model of music description that has learned everything it knows by reading reviews and listening to the targets.

Of course, reviews have their problems. By their nature they are hardly objective – the author's own background and musical knowledge color each review. As Figure 2 illustrates, music reviews can often be cluttered with 'outside-world' information, such as personal relationships and celebrity trivia. While these non-musical tidbits are entertaining for the reader and sometimes (if



For the majority of Americans, it's a given: summer is the best season of the year. Or so you'd think, judging from the anonymous TV ad men and women who proclaim, "Summer is here! Get your [insert iced drink here] now!"-- whereas in the winter, they regret to inform us that it's time to brace ourselves with a new Burlington coat. And TV is just an exaggerated reflection of ourselves; the hordes of convertibles making the weekend pilgrimage to the nearest beach are proof enough. Vitamin D overdoses abound. If my tone isn't suggestive enough, then I'll say it flat out: I hate the

Beginning with "Caring Is Creepy," which opens this album with a psychedelic flourish that would not be out of place on a late-1960s Moody Blues, Beach Boys, or Love release, the Shins present a collection of retro pop nuggets that distill the finer aspects of classic acid rock with surrealistic lyrics, independently melodic bass lines, jangly guitars, echo laden vocals, minimalist keyboard motifs, and a myriad of cosmic sound effects. With only two of the cuts clocking in at over four minutes, *Oh Inverted World* avoids the penchant for self-indulgence that befalls most outfits who worship at the

Figure 2. The first few lines of two separate reviews of the same album (The Shins' "Oh Inverted Word.") Top: Ryan Kearney, Pitchforkmedia.com. Bottom: Tom Semioli, All Music Guide.

obliquely) give a larger picture of the music in question, our current purpose would be best served by more concise reviews that concentrated on the contents of the album so that our models of music understanding and similarity are dealing with purely content-related features.

In this paper we study a large corpus of music audio and corresponding reviews as an exploratory work into the utility of music reviews for retrieval tasks. We are specifically interested in the problems of similarity and recommendation, and view the review parsing and term grounding work in this paper as a necessary step to gathering the knowledge required to approximate human musical intelligence. For example, by limiting reviews to 'musically salient' terms grounded by our learning system, a community-opinion model of similarity, based only on text, can be built with high accuracy.

We first present a computational representation of parsing for descriptive text and an audio representation that captures different levels of musical structure. We then show methods for linking the two together, first to create models for each term that can be evaluated, but also to cull non-musical and biased information from reviews. We also show results in classifying the author's overall opinion of the work, as expressed in symbolic "star-rating" attributes provided by the review, by learning the relationship between the music and its fitness score. Putting these approaches together opens the door to an on-line "automatic record review" that can classify new music with numerous human-readable and understandable labels. These labels can be used directly in an interface or used as inputs to subsequent similarity, classification or recommendation systems.

2. BACKGROUND

Our work has concentrated on extracting meaning from music, using language processing and data mining techniques to uncover connections between the perception (audio stream) and description. Many interesting results have arisen from this work, including models of metadata derived from musical communities [2], a "query by description" system that allows users a natural interface for music retrieval [3], and a new method of *semantic rank reduction* where the observations are de-correlated based on meaning rather than purely statistics [4]. By associating listener reactions to music (observed through many mechanisms from player logs through to published reviews) with analyses of the audio signal, we can automatically infer novel relations on new, "unheard" music. This paper ties some of these threads together for a an approach to extracting reliable, consensus information from disparate online reviews.

2.1. Related Work

2.1.1. Music Analysis

Systems can understand music enough to classify it by genre, style, or nationality, as long as the systems are trained with hand-labeled data e.g. [5, 6]. The link between musical content and generalized descriptive language is not as prominent, although [7] shows that certain style-related terms such as 'lyrical' or 'frantic' can be learned from the score level.

2.1.2. Grounding

In the domain of general audio, recent work has linked sound samples to description using the labeled descriptions on the sample sets [8]. In the visual domain, some work has been undertaken attempting to learn a link between language and multimedia. The lexicon-learning aspects in [9] study a set of fixed words applied to an image database and use a method similar to EM (expectation-maximization) to discover where in the image the terms (nouns) appear; [10] outlines similar work. Regier has studied the visual grounding of spatial terms across languages, finding subtle effects that depend on the relative shape, size, and orientation of objects [11]. In [15], aspects of learning shape and color terms were explored along with some of the first steps in perceptually-grounded grammar acquisition.

3. THE “MIT AUDIO+AUDIENCE” TESTBED

The set of music used in this article and elsewhere is based on the Minnowmatch testbed [17] extended with a larger variety of music (instead of just pop) by removing the popularity constraint. (Minnowmatch’s music was culled from the top 1,000 albums on a peer-to-peer network.) We have also added a regularized set of cultural metadata for each artist, album, and song. In this paper we report results on a set of 600 albums from roughly 500 artists. Each artist has concordant community metadata vectors [2] and each album has at least two reviews, one from the metadata provider All Music Guide [18] (AMG) and one from the popular record review and music culture web site Pitchfork Media [19] (Pitchfork). Most records also have tagged community reviews from other sources, such as on-line record stores. Other sources of community information in this testbed include usage data and artist similarity results from the Musicseer [20] survey.

4. READING THE REVIEWS

There are innumerable ways of representing textual information for machine understanding, and in our work we choose the simplest and most proven method of frequency counting. Reviews are in general short (one to three paragraphs), are always connected to the topic (although not always directly) and do not require special parsing or domain-specific tools to encode. In our recent work we used a very general model of *community metadata* [2] which creates a machine understandable representation of artist description by searching the Internet for the artist name and performing natural language processing on the retrieved pages. Since those results were naturally noisier (all text on a web page vs. a succinct set of three paragraphs) we needed various post-crawling processing tricks to clean up the data. In this experiment we borrow tools and ideas from the community metadata crawler but mostly rely on simple information retrieval techniques.

The reviews were downloaded using a specialized crawler and added to the Audio+Audience testbed. We chose 600 albums, two reviews for each (AMG and Pitchfork) to use later in interrater studies and as an agreement measure. All markup was removed and each review is split into plaintext sentences. We decompose the reviews into n-grams (terms of word length n), adjective sets (using a part-of-speech tagger [21]) and noun phrases (using a lexical chunker [22]). We compute the term frequency of each term as it occurs in a review i.e. if there were 50 adjectives in a review of an album, and *loud* appeared five times, *loud*’s tf is 0.1.We then compute global document frequencies (if *loud* occurred in 30 of the 600 reviews, its df would be 0.05).

Each pair $\{review, term\}$ retrieved is given an associated salience weight, which indicates the relative importance of $term$ as associated to the $review$. These saliences are computed using the TF-IDF measure; simply tf/df. The intuition behind TF-IDF is to reward words that occur frequently in a topic but not overall. For example, the term *guitars* might have a high tf for a rock review but also has a high df in general; this downweights it. But *electric banjo* has a high tf for particular reviews and a low df, which causes it to have a high salience weight. We limit the $\{review, term\}$ pairs to terms that occur in at least three reviews so that our machine learning task is not overwhelmed with negative bias. See Table 1 for example top-scoring salience terms. We make use of these TF-IDF salience scores as a metric to only allow certain terms to be considered by the machine learning systems that learn the relationship between terms and audio. We limit terms by their df overall and then limit ‘learnable’ terms by their specific TF-IDF per album review. Previous work [3] directly used the TF-IDF scores as a regression target in the learning system; we found this to lessen accuracy as TF-IDF does not have good normalizing metric.

We also parse the explicit ratings of each album in our collection. Pitchfork rates each record on a (0..10) scale with decimals (for 100 steps), while AMG uses a star system that has 9 distinct granulations.

Our choice of AMG and Pitchfork as our review sources was not accidental: we selected them as two opposite poles in the music criticism space. AMG is a heavily edited metadata source whose reviews are consistently concise, short and informational. Pitchfork’s content is geared towards a younger audience and more ‘buzz-friendly’ music, acting as more of a news site than a review library. The tone of the latter is very informal and not very consistent. This makes Pitchfork our self-selected ‘worst case scenario’ for ground truth as our results later show– the ratings and reviews have little representation in the audio itself. Likewise, AMG acts as a best case and our systems have an easier time linking their descriptions and ratings to music. Nonetheless the two sources serve to complement each other. There is much to music outside the signal, and the culture and buzz extracted from Pitchfork’s reviews could be extracted and represented for other purposes.

5. LISTENING TO THE MUSIC

5.1. The “Penny” Cepstral Features

A number of subproblems arise when attempting to discover arbitrary lexical relations between words and music. The foremost problem is one of scale: any lexical unit attached to music can agree with the entire artist (long-term scale), just an album, just a song or piece, or perhaps a small part of the song. Even lower-level are relations between descriptions and instruments, or filters or tones (“This sound is dark,” or “these guitars are grating.”) The problems are further exacerbated when most machine learning systems treat observations as unordered frames.

We are looking for a model of auditory perception that attempts to simultaneously capture many levels of structure within a musical segment, but does so without experimenter bias or supervision guidance. A common downfall

Hrvatski noun phrases	adjectives	Richard Davies noun phrases	adjectives	Hefner noun phrases	adjectives
swarm & dither	processed	telegraph	creatively	hefner	disenchanted
hrvatksi	feedback	richard davies	unsuccessful	the indie rock trio	contemptuous
synth patch	composed	eric matthews and cardinal	instrumentalist	such a distant memory	fashionable
baroque symphonies	glitch	the moles	australian	an emo band	beloved
old-fashioned human emotion	psychotic	poetic lyrics	quieter	guitars and pianos	puzzling
polyrhythmic pandemonium	cheerful	the kinks surface	terrific	some humor	nasal
his breaks fascination	crazed	his most impressive record	reflective	singer darren hayman	ugly

Table 1. Selected top-scoring noun phrase and adjective terms (TFIDF) from three combined record reviews.

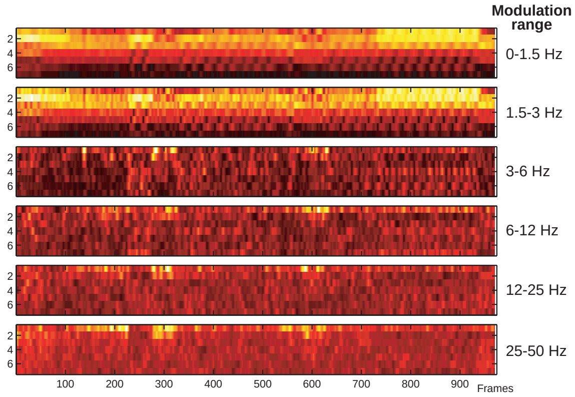


Figure 3. The "Penny" cepstral features for generalized semantic analysis of audio. Six levels of structure are decoded for this song ("A Journey to Reedham" by Squarepusher), corresponding to different ranges of modulation frequencies.

of many heuristically musical feature encodings is their reliance on the observation being "cleanly musical" – for example, a pitch and beat based feature encoding does not generalize well to non-tonal music or freeform pieces. We would also like our learning algorithm to be able to handle generalized sound.

Our previous work [3] uses a very low-level feature of audio, the power spectral density (PSD) at 512 points. Roughly, a PSD is the mean of STFT bins over some period of time (we used 4 seconds in our work). While our results were encouraging, we ran up against problems of scale in trying to increase our generalization power. As well, we were not capturing time-dependent information such as "faster" or "driving." We also attempted to use the MPEG-7 time-aware state path representation of audio proposed in [23] which gave us perceptibly more "musical" results but still did not allow for varying levels of musical structure.

Our new feature space, nicknamed "Penny" is based on the well known Mel-frequency Cepstral Coefficients (MFCCs) from speech recognition. We take MFCCs at a 100 Hz sample rate, returning a vector of 13 bins per audio frame. We then stack successive time samples for each MFCC bin into 64 point vectors and take a second Fourier transform on these per-dimension temporal energy envelopes. We aggregate these results into 6 octave wide bins to create a "modulation spectrum" showing the dominant scales of energy variation for each cepstral component over a range of 1.5 Hz to 50 Hz. The result is six matrices (one for each modulation spectrum octave) each containing 13 bins of cepstral information, sampled at, for instance, 10 Hz (to give roughly 70% overlap between successive modulation spectral frames). The first matrix gives information about slow variations in the cepstral magnitudes, indicating things like song structure or large changes in the piece, and each subsequent matrix concentrates on higher frequencies of modulation for each cepstral coefficient. An example set of six matrices from the Penny analysis can be seen in Figure 3.

6. LEARNING THE LANGUAGE

In this section we discuss the machinery to learn the relation between the audio features and review text. The approach we use is related to our previous work, where we pose the problem as a multi-class classification problem. In training, each audio feature is associated with some salience weight of each of the 5,000 possible terms that our review crawler discovered. Many of these classes are unimportant (as in the case of terms such as 'talented' or 'cool'– meaningless to the audio domain). We next show our attempt at solving these sorts of problems using a classifier technique based on support vector machines [24].

6.1. Regularized Least-Squares Classification

Regularized Least-Squares Classification [25] requires solving a single system of linear equations after embedding the data in a kernel space. Recent work [26, 25] has shown that the accuracy of RLSC is essentially identical to that of the closely related support vector machine, but at a fraction of the computational cost. We arrange our audio observations in a kernel-space gram matrix K, where $K_{ij} \equiv K_f(x_i, x_j)$, a generalized dot product between x_i and x_j. Thus, if the generalized dot product is considered a similarity function, the gram matrix compares each point against every other in the example space. We usually use the Gaussian kernel,

$$K_f(x_1, x_2) = e^{-\frac{(|x_1 - x_2|)^2}{\sigma^2}} \tag{1}$$

where $|x - y|$ is the conventional Euclidean distance between two points, and σ is a parameter we keep at 0.5.

Training an RLSC system consists of solving the system of linear equations

$$(K + \frac{I}{C})\mathbf{c} = \mathbf{y}, \qquad (2)$$

where K is the kernel matrix, $\mathbf{c}$ is a classifier 'machine,' $\mathbf{y}$ is the truth value, and C is a user-supplied *regularization constant* which we keep at 10.[1] The crucial property of RLSC for this task is that if we store the inverse matrix $(K + \frac{I}{C})^{-1}$, then for a new right-hand side $\mathbf{y}$ (i.e. a new set of truth term values we are trying to predict), we can compute the new classifier $\mathbf{c}$ via a simple matrix multiplication. Thus, RLSC is very well-suited to problems of this scale with a fixed set of training observations and a large number of target classes, some of which might be defined after the initial analysis of the training points.

To compute a set of term classifiers for audio observations (i.e. given an audio frame, which terms are associated and with what magnitude?) we form a kernel-space gram matrix from our Penny features, add the regularization constant, and invert. We then multiply the resultant matrix by a set of 'term truth vectors' derived from the training data. These are vectors with one value for each of the examples in the training kernel matrix, representing the salience (from the TF-IDF computation) of that term to that audio frame.[2] This multiplication creates a 'machine' $\mathbf{c}$ which can then be applied to the test examples for evaluation.

7. EXPERIMENTS

We conducted a set of experiments, first testing our feature extraction and learning algorithms' capability to generalize a review for a new piece of music, then using the precision of each term model to cull non-musical (ungroundable) phrases and sentences from reviews, and lastly trying to learn the relationship between audio and review rating. Each task runs up against the problem of ground truth: our models are trained to predict very subjective information described only through our own data. We discuss each experiment below with directions into future work.

7.1. Learning Results

To generate reviews automatically from audio we must first learn a model of the audio-to-term relations. We extract textual features from reviews for noun phrase and adjective types as above and then compute the Penny feature space on our set of 600 albums, choosing four songs at random from each. (We start with MP3 audio and convert to mono and downsample to 11 kHz.) We use the lowest two modulation frequency bins of the Penny feature across all cepstra for a feature dimension of 26. We use a 10 Hz feature framerate that is then downsampled to 1 Hz. We split the albums into testing and training, with half of the albums in each. Using the RLSC method described above we compute the gram matrix on the training data and then invert, creating a new $\mathbf{c}$ for each term in our review corpus.

7.2. Evaluation of Predicted Terms

To evaluate the models on new albums we compute the testing gram matrix and check each learned $\mathbf{c}$ against each audio frame in the test set.

We used two separate evaluation techniques to show the strength of our term predictions. One metric is to measure classifier performance with the recall product $P(a)$: if $P(a_p)$ is the overall positive accuracy (i.e. given an audio frame, the probability that a positive association to a term is predicted) and $P(a_n)$ indicates overall negative accuracy, $P(a)$ is defined as $P(a_p)P(a_n)$. This measure gives us a tangible feeling for how our term models are working against the held out test set and is useful for grounded term prediction and the review trimming experiment below. However, to rigorously evaluate our term model's performance in a review generation task, we note that this value has an undesirable dependence on the prior probability of each label and rewards term classifiers with a very high natural df, often by chance. Instead, for this task we use a model of relative entropy, using the Kullback-Leibler (K-L) distance to a random-guess probability distribution.

We use the K-L distance in a two-class problem described by the four trial counts in a confusion matrix:

	"funky"	**"not funky"**
funky	a	b
not funky	c	d

a indicates the number of frames in which a term classifier positively agrees with the truth value (both classifier and truth say a frame is 'funky,' for example). b indicates the number of frames in which the term classifier indicates a negative term association but the truth value indicates a positive association (the classifier says a frame is not 'funky,' but truth says it is). The value c is the amount of frames the term classifier predicts a positive association but the truth is negative, and the value of d is the amount of frames the term classifier and truth agree to be a negative association. We wish to maximize a and d as correct classifications; by contrast, random guessing by the classifier would give the same ratio of classifier labels regardless of ground truth i.e. $a/b \approx c/d$. With $N = a + b + c + d$, the K-L distance between the observed distribution and such

[1] We arrived at 0.5 for σ and 10 for C after experimenting with the Penny features' performance on an artist identification task, a similar music-IR problem with better ground truth.

[2] We treat all audio frames derived from an album the same in this manner. If a review claims that "The third track is slow and plodding" this causes every frame of audio derived from that album to be considered slow and plodding.

random guessing is:

$$KL = \frac{a}{N}\mathbf{log}\left(\frac{N\,a}{(a+b)\,(a+c)}\right) + \frac{b}{N}\mathbf{log}\left(\frac{N\,b}{(a+b)\,(b+d)}\right) + \frac{c}{N}\mathbf{log}\left(\frac{N\,c}{(a+c)\,(c+d)}\right) + \frac{d}{N}\mathbf{log}\left(\frac{N\,d}{(b+d)\,(c+d)}\right) \tag{3}$$

This measures the distance of the classifier away from a degenerate distribution; we note that it is also the mutual information (in bits, if the logs are taken in base 2) between the classifier outputs and the ground truth labels they attempt to predict.

Table 2 gives a selected list of well-performing term models. Given the difficulty of the task we are encouraged by the results. Not only do the results give us term models for audio, they also give us insight into which terms and description work better for music understanding. These terms give us high semantic leverage without experimenter bias: the terms and performance were chosen automatically instead of from a list of genres.

7.3. Automatic review generation

The multiplication of the term model **c** against the testing gram matrix returns a single value indicating that term's relevance to each time frame. This can be used in review generation as a confidence metric, perhaps setting a threshold to only allow high confidence terms. The vector of term and confidence values for a piece of audio can also be fed into other similarity and learning tasks, or even a natural language generation system: one unexplored possibility for review generation is to borrow fully-formed sentences from actual reviews that use some amount of terms predicted by the term models and form coherent paragraphs of reviews from this generic source data. Work in language generation and summarization is outside the scope of this article but the results for the term prediction task and the below review trimming task are promising for these future directions.

One major caveat of our review learning model is its time insensitivity. Although the feature space embeds time at different levels, there is no model of intra-song changes of term description (a loud song getting soft, for example) and each frame in an album is labeled the same during training. We are currently working on better models of time representation in the learning task. Unfortunately, the ground truth in the task is only at the album level and we are also considering techniques to learn finer-grained models from a large set of broad ones.

adj Term	K-L bits	np Term	K-L bits
aggressive	0.0034	reverb	0.0064
softer	0.0030	the noise	0.0051
synthetic	0.0029	new wave	0.0039
punk	0.0024	elvis costello	0.0036
sleepy	0.0022	the mud	0.0032
funky	0.0020	his guitar	0.0029
noisy	0.0020	guitar bass and drums	0.0027
angular	0.0016	instrumentals	0.0021
acoustic	0.0015	melancholy	0.0020
romantic	0.0014	three chords	0.0019

Table 2. Selected top-performing models of adjective and noun phrase terms used to predict new reviews of music with their corresponding bits of information from the K-L distance measure.

7.4. Review Regularization

Many problems of non-musical text and opinion or personal terms get in the way of full review understanding. A similarity measure trained on the frequencies of terms in a user-submitted review would likely be tripped up by obviously biased statements like "This record is awful" or "My mother loves this album." We look to the success of our grounded term models for insights into the *musicality* of description and develop a 'review trimming' system that summarizes reviews and retains only the most descriptive content. The trimmed reviews can then be fed into further textual understanding systems or read directly by the listener.

To trim a review we create a grounding sum term operated on a sentence s of word length n,

$$g(s) = \frac{\sum_{i=0}^{n} P(a^i)}{n} \tag{4}$$

where a perfectly grounded sentence (in which the predictive qualities of each term on new music has 100% precision) is 100%. This upper bound is virtually impossible in a grammatically correct sentence, and we usually see $g(s)$ of {0.1% .. 10%}. The user sets a threshold and the system simply removes sentences under the threshold. See Table 3 for example sentences and their $g(s)$. We see that the rate of sentence recall (how much of the review is kept) varies widely between the two review sources; AMG's reviews have naturally more musical content. See Figure 4 for recall rates at different thresholds of $g(s)$.

7.5. Rating Regression

Lastly we consider the explicit rating categories provided in the review to see if they can be related directly to the audio, or indeed to each other. Our first intuition is that learning a numerical rating from audio is a fruitless task as the ratings frequently reflect more information from outside the signal than anything observable in the waveforms. The public's perception of music will change, and as a result reviews of a record made only a few months apart might wildly differ. In Figure 1 we see that correlation of ratings between AMG and Pitchfork is generally low

Sentence	$g(s)$
The drums that kick in midway are also decidedly more similar to Air's previous work.	3.170%
But at first, it's all Beck: a harmonica solo, folky acoustic strumming, Beck's distinctive, marble-mouthed vocals, and tolls ringing in the background.	2.257%
But with lines such as, ”We need to use envelope filters/ To say how we feel,” the track is also an oddly beautiful lament.	2.186%
The beat, meanwhile, is cut from the exact same mold as The Virgin Suicides– from the dark, ambling pace all the way down to the angelic voices coalescing in the background.	1.361%
After listing off his feelings, the male computerized voice receives an abrupt retort from a female computerized voice: ”Well, I really think you should quit smoking.”	0.584%
I wouldn't say she was a lost cause, but my girlfriend needed a music doctor like I needed, well, a girlfriend.	0.449%
She's taken to the Pixies, and I've taken to, um, lots of sex.	0.304%
Needless to say, we became well acquainted with the album, which both of us were already fond of to begin with.	0.298%

Table 3. Selected sentences and their $g(s)$ in a review trimming experiment. From Pitchfork's review of Air's "10,000 Hz Legend."

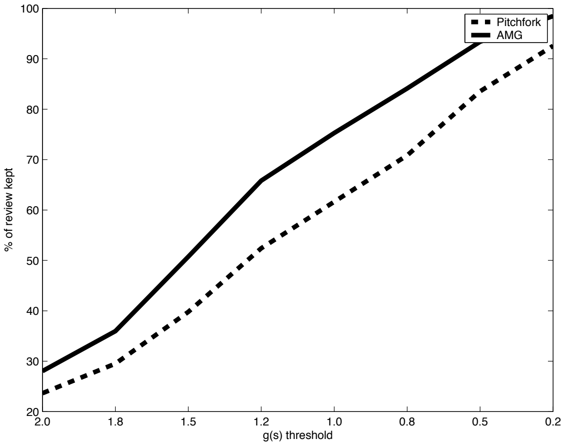


Figure 4. Review recall rates at different $g(s)$ thresholds.

with a correlation coefficient of $r = 0.264$ (where a random pairing of ratings over multiple simulations gives us a coefficient of 0.071 with 95% confidence.)

Although we assume there is no single overall set of record ratings that would satisfy both communities, we do believe AMG and Pitchfork represent two distinct sets of "collective opinion" that might be successfully modeled one at a time. A user model might indicate which community they 'trust' more, and significance could then be extracted only from that community. The experiment then becomes a test to learn each reviewing community's ratings, and to see if each site maintains consistency in their scores.

We use our Penny features again computed on frames of audio derived from the albums in the same manner as our review learning experiment. We treat the problem as a multi-dimensional regression model, and we use a support vector machine classifier to perform the regression. We use the same album split for testing and training as above, and train each frame of audio against the rating (scaled to 0..1). We then evaluate the model against the test set and compute the correlation coefficient against the actual rating. The AMG model did well with a correlation coefficient of $r = 0.147$. Through empirical simulation we established that a random association of these two datasets gives a correlation coefficient of magnitude smaller than $r = 0.080$ with 95% confidence. Thus, these results indicate a very significant correlation between the automatic and ground-truth ratings.

The Pitchfork model did not fare as well with $r = 0.127$ (baseline of $r = 0.082$ with 95% confidence.) Figure 1 shows the scatter plot/histograms for each experiment; we see that the audio predictions are mainly bunched around the mean of the ground truth ratings and have a much smaller variance. Visually, it is hard to judge how well the review information has been captured. However, the correlation values demonstrate that the automatic analysis is indeed finding and exploiting informative features.

While our results in the rating regression experiment were less than excellent we consider better community modeling part of future work. Within a community of music listeners the correlation of opinions of albums will be higher and we could identify and tune models to each community.

8. CONCLUSIONS

We are using reviews and general text descriptions, much as human listeners do, to move beyond the impoverished labels of genres and styles which are ill-defined and not generalizable. Human description is a far richer source of target classes and clusters than marketing tags which can have almost no relationship to audio content. By identifying communities of music preference and then learning the language of music we hope to build scalable models of music understanding. Review analysis represents one source of information for such systems, and in this article we have shown analysis frameworks and results on learning the crucial relation between review texts and the music they describe.

9. ACKNOWLEDGEMENTS

We are very grateful for the help of Ryan McKinley (Computing Culture Group, MIT Media Lab) in arranging the music and metadata testbed.

10. REFERENCES

[1] K. Dave, S. Lawrence, and D. Pennock, “Mining the peanut gallery: Opinion extraction and semantic classification of product reviews,” in *International World Wide Web Conference*, Budapest, Hungary, May 20–24 2003, pp. 519–528.

[2] B. Whitman and S. Lawrence, “Inferring descriptions and similarity for music from community metadata,” in *Proc. Int. Computer Music Conference 2002 (ICMC)*, September 2002, pp. 591–598.

[3] B. Whitman and R. Rifkin, “Musical query-by-description as a multi-class learning problem,” in *Proc. IEEE Multimedia Signal Processing Conference (MMSP)*, December 2002.

[4] B. Whitman, “Semantic rank reduction of music audio,” in *Proc. IEEE Worksh. on Apps. of Sig. Proc. to Acous. and Audio*, 2003.

[5] G. Tzanetakis, G. Essl, and P. Cook, “Automatic musical genre classification of audio signals,” 2001. [Online]. Available: citeseer.nj.nec.com/tzanetakis01automatic.html

[6] W. Chai and B. Vercoe, “Folk music classification using hidden markov models,” in *Proc. International Conference on Artificial Intelligence*, 2001.

[7] R. B. Dannenberg, B. Thom, and D. Watson, “A machine learning approach to musical style recognition,” in *In Proc. 1997 International Computer Music Conference*. International Computer Music Association., 1997, pp. 344–347. [Online]. Available: citeseer.nj.nec.com/dannenberg97machine.html

[8] M. Slaney, “Semantic-audio retrieval,” in *Proc. 2002 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2002.

[9] P. Duygulu, K. Barnard, J. D. Freitas, and D. Forsyth, “Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary,” 2002. [Online]. Available: citeseer.nj.nec.com/duygulu02object.html

[10] K. Barnard and D. Forsyth, “Learning the semantics of words and pictures,” 2000. [Online]. Available: citeseer.nj.nec.com/barnard00learning.html

[11] T. Regier, *The human semantic potential*. Cambridge, MA: MIT Press, 1996.

[12] D. Bailey, “When push comes to shove: A computational model of the role of motor control in the acquisition of action verbs,” Ph.D. dissertation, University of California at Berkeley, 1997.

[13] S. Narayanan, “Knowledge-based action representations for metaphor and aspect (karma),” Ph.D. dissertation, University of California at Berkeley, 1997.

[14] J. Siskind, “Grounding the Lexical Semantics of Verbs in Visual Perception using Force Dynamics and Event Logic,” *Journal of Artificial Intelligence Research*, vol. 15, pp. 31–90, 2001.

[15] D. Roy, “Learning words from sights and sounds: A computational model,” Ph.D. dissertation, Massachusetts Institute of Technology, 1999.

[16] D. Cruse, *Lexical Semantics*. Cambridge University Press, 1986.

[17] B. Whitman, G. Flake, and S. Lawrence, “Artist detection in music with minnowmatch,” in *Proc. 2001 IEEE Workshop on Neural Networks for Signal Processing*, Falmouth, Massachusetts, September 10–12 2001, pp. 559–568.

[18] “All music guide.” [Online]. Available: http://www.allmusic.com

[19] “Pitchfork media.” [Online]. Available: http://www.pitchforkmedia.com

[20] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence, “The quest for ground truth in musical artist similarity,” in *Proc. International Symposium on Music Information Retrieval ISMIR-2002*, 2002.

[21] E. Brill, “A simple rule-based part-of-speech tagger,” in *Proc. ANLP-92, 3rd Conference on Applied Natural Language Processing*, Trento, IT, 1992, pp. 152–155. [Online]. Available: citeseer.nj.nec.com/article/brill92simple.html

[22] L. Ramshaw and M. Marcus, “Text chunking using transformation-based learning,” in *Proc. Third Workshop on Very Large Corpora*, D. Yarovsky and K. Church, Eds. Somerset, New Jersey: Association for Computational Linguistics, 1995, pp. 82–94. [Online]. Available: citeseer.nj.nec.com/ramshaw95text.html

[23] M. Casey, “General sound recognition and similarity tools,” in *MPEG-7 Audio Workshop W-6 at the AES 110th Convention*, May 2001.

[24] V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.

[25] R. M. Rifkin, “Everything old is new again: A fresh look at historical approaches to machine learning,” Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[26] G. Fung and O. L. Mangasarian, “Proximal support vector classifiers,” in *Proc. Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Provost and Srikant, Eds. ACM, 2001, pp. 77–86.