SEMANTIC RANK REDUCTION OF MUSIC AUDIO

Brian Whitman

MIT Media Lab
20 Ames St. E15-491
Cambridge, MA 02139 USA
bwhitman@media.mit.edu

ABSTRACT

Audio understanding and classification tasks are often aided by a reduced dimensionality representation of the source observations. For example, a supervised learning system trained to detect the genre or artist of a piece of music performs better if the input nodes are statistically de-correlated, either to prevent overfitting in the learning process or to 'anchor' similar observations to cluster centroids in the observation space. We provide an alternate approach that decomposes audio observations of music into *semantically significant* dimensions where each resultant dimension corresponds to the perceived meaning of the audio, and only the most significant meanings (those which are most effective in describing music audio) are kept. We show a fundamentally unsupervised method to automatically obtain this decomposition and compare its performance in a music understanding task against statistical de-correlation approaches such as PCA and non-negative matrix factorization (NMF).

1. INTRODUCTION

Music is unlike most audio due to its strong descriptive composition. Any given artist will have thousands of pages of description, opinion, and cultural backstory which is usually integral to fully understanding the content. However, most music retrieval and classification systems treat music audio as statistical observations, ignoring any sense of its *meaning*– the oftentimes cultural component of music that separates it from sound. We aim to connect current statistical de-correlation techniques currently helpful in increasing the accuracy of music classifiers with the semantics of music, in effect creating a set of 'semantic basis functions' that can decompose new music audio into a compact representation that retains a maximal link from meaning to perception.

Our system automatically generates an ordered list of audio observation-to-term classifiers from internet description of artists that can be used for future decomposition of music audio. We show in this paper that these functions retain more information about the underlying music than other popular statistical de-correlation approaches evaluated in a music understanding task. As an added benefit, they provide human-readable labels on each weight for future use in clean interfaces to music retrieval.

2. BACKGROUND

This work is based on current research in linking meaning to perception in music. In [1] we evaluate such a system in a query-by-description task, linking adjective terms collated from the internet against a frame-based audio representation. In [2] we extend the model to parameter spaces such as "loud...quiet" and "high...low." Similar work in audio such as [3] takes a more supervised approach, learning hierarchies of description from labeled descriptions of sound samples.

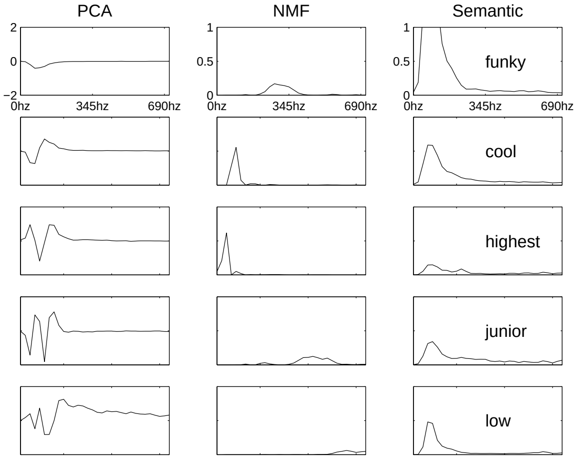

Figure 1: Comparison of the top five bases for each type of decomposition, trained from a set of five second power spectral density frames. The PCA weights aim to maximize variance, the NMF weights try to find separable additive parts, and the semantic weights map the best possible labels to the generalized observations.

Related work in [4] performs a similar semantic decomposition of music audio, using genre terms provided by the artist plus two terms provided by the authors as 'anchors'. They found better results with the anchors in place on an artist detection task. We provide here a view of anchor models that are automatically derived with the maximal semantic attachment in place. As well, we look towards simpler, more primitive labels such as 'loud' and 'funky' rather than complex (and often marketing-influenced) genre tags like 'Rock/Pop.'

We compare our semantic decomposition against commonly-used statistical approaches to rank reduction such as principal components analysis (PCA) (based on the singular value decomposition [5]) and non-negative matrix factorization (NMF) [6]. NMF performs a similar decomposition as PCA but constrains its bases to be positive in an attempt to mimic "part-finding" in observations. We find that noisy audio observations fare better with PCA, but highly harmonic musical content (such as piano solo pieces)

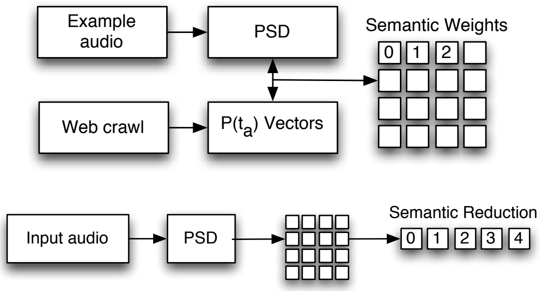

Figure 2: Top: obtaining "semantic basis functions" by learning the relations between the spectral properties of a music signal and the adjective vectors of descriptions used for that music. The result is an ordered list of semantic weights (with labels), each one linked stronger in meaning than the next. Bottom: applying the stored weights to new input audio to obtain a dimensionality reduced representation of the audio that retains maximal semantic content.

are a good fit for the additive nature of NMF.

3. OBTAINING A SEMANTIC DECOMPOSITION

We find our semantic decomposition by discovering the relationships between the input audio observations and descriptions of the audio found on the internet. We use the Klepmit "community metadata" system (described in [7] and used in [8] as an artist similarity dataset) to obtain a description vector of each artist represented in an input set of music, and retain only the adjective vector. In Klepmit, a 'description vector' is a list of terms associated with an artist and the probability $P(t_a)$ of that term being relevant to the artist. See Table 1 for an example vector. We note that by the unsupervised and open nature of the term collection, a lot of noise (opinions, non-music related data, technical terms) comes through in the description vector. But we rely on the semantic attachment step to make sure terms are adequately related to the audio observations.

We choose a set of music audio, split it into equal-sized train and test segments, and label the observations with the artist names. The Klepmit system retrieves the term types from the artist names[1] and sets up the $P(t_a)$ vectors for each artist a and term t. (We usually end up with 1000 to 2000 unique adjective t.) Concurrently, we form the audio from each artist into a frame-based representation. In our work we use the power spectral density estimate (PSD) over each 5s of audio. We then feed the training audio observations and the description vectors to a multiclass learning system to learn a new description vector for incoming audio frames.

3.1. Regularized Least-Squares Classification

Regularized Least-Squares Classification [9] allows us to solve multi-class problems where there are a large number of target classes and a fixed set of source observations. It is related to the Support Vector Machine [10] in that they are both instances of Tikhonov regularization [11], but whereas training a Support Vector Machine requires the solution of a constrained quadratic programming problem, training RLSC only requires solving a single system of linear equations. Recent work [12], [9] has shown that the accuracy of RLSC is essentially identical to that of SVMs.

We arrange our audio observations in a Gram matrix K, where $K_{ij} \equiv K_f(x_i, x_j)$ using the *kernel function* K_f. $K_f(x_1, x_2)$ is a generalized dot product (in a Reproducing Kernel Hilbert Space [13]) between x_i and x_j. It is sometimes helpful to think of the generalized dot product as a similarity function, comparing each point against the other in your example space. For audio-derived observations as in the PSD we usually use the Gaussian kernel

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

where σ is a parameter we keep at 0.5.

Then, training an RLSC system consists of solving the system of linear equations

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

where K is the distance 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. The resulting real-valued classification function f is

$$f(x) = \sum_{i=1}^{\ell} c_i K(x, x_i). \tag{3}$$

The crucial property of RLSC is that if we store the inverse matrix $(K + \frac{I}{C})^{-1}$, then for a new right-hand side $\mathbf{y}$ (for a new set of truth values we are trying to generalize to), we can compute the new classifier $\mathbf{c}$ via a simple matrix multiplication.

3.2. Ordering Semantic Labels

We have the RLSC process create a $\mathbf{c_t}$ term classifier for each descriptor t in our crawl. To do so, we arrange a new $\mathbf{y_t}$ composed of the Klepmit-derived scores for each descriptor on the fly. (For example, $\mathbf{y}_{funky}$ is a vector of the amount of 'funky' for each audio frame.) To determine which terms have stronger links between meaning and perception than others, we evaluate each $\mathbf{c_t}$ against the test set of audio. This allows us to measure how well each term does in finding a relation between meaning and perception. High accuracy classifiers usually link to 'musical' terms, such as "loud," "funky," "electronic," or "acoustic" – and low scoring classifiers link to cultural or noise terms such as "untalented" or "sexy." Our previous work [2] discusses this separation in more detail. To formalize performance of audio-to-term classifiers, we compute the weighted performance $P(t)$ of each $\mathbf{c_t}$ on the test set. Where $P(t_p)$ indicates overall positive accuracy (for example, given an audio-derived observation labeled 'funky' by Klepmit, the probability that the audio classifier agrees) and $P(t_n)$ indicates overall negative accuracy (that the classifier and Klepmit agree not to label a frame 'funky'), $P(t)$ is defined as $P(t_p)P(t_n)$, which should remain significant even in the face of extreme negative output class bias.

We sort the term list by $P(t)$, and leave it up to the user to select a rank r. The semantic basis functions are defined as the top r $\mathbf{c_t}$ classifiers ordered by our sort. (See Figure 1 for PSD bases of the top five classifiers kept in our experiment.) New data can be parameterized by a set of r coefficients, each one the result of asking the top audio-to-term classifiers to return a scalar of what

[1] Klepmit parses all text found near an artist name anywhere on the web. For more information, see [7]

adj Term	$P(t_a)$	adj Term	$P(t_a)$
cynical	0.2997	produced	0.1143
smooth	0.0792	dark	0.0583
particular	0.0571	loud	0.0558
amazing	0.0457	vocal	0.0391
unique	0.0362	simple	0.0354

Table 1: Small subset of an example description vector.

they think of the incoming audio observation. This parameterization aims to retain maximal semantic value, where each dimension corresponds to some high-level descriptor of the input perception.

This model closely follows recent work in 'categorization by combining' [14] or 'anchor models' [4] where a series of subclassifier "experts" each feed into a larger combiner classifier. But we note a crucial difference in that our experts that serve to cluster the input observations into a reduced rank for further classification are completely autonomously learned, with no experimenter bias on possible cluster points or music content. The only user intervention necessary is to label the music audio with its artist, a process that can be automated from most metadata sources. As well, we can claim that our sorted list of experts expresses the highest semantic content possible given the chosen r due to its evaluation performance on the test audio, whereas more supervised methods are only generalizable to the experimenter's view of the problem.

3.3. Application of Semantic Rank-Reduction

The set of $\mathbf{c_t}$ can be stored away for future use against any new set of music given that the representation of audio remains the same. For example, a generalized semantic rank reduction set of classifiers can be learned from a large set of all possible genres of music audio and later used against a new set of music audio. In this application case, the new set of audio does not need to be labeled with an artist tag or with description and we can view the semantic rank reduction of this data as an analogy to applying a weighting transform learned from a previous PCA. We note that some of the same caveats apply: your bases should be learned from data that will be similar to data found in your classification task. Semantic classifiers trained on only classical music, for example, might retrieve specific term relations (such as "bright" or "brassy") and will not generalize well to rap music.

4. EXPERIMENTS AND RESULTS

4.1. Artist Classification

We use an artist identification problem to evaluate different dimensionality reduction methods. Artist ID [15] [16] is a well-defined problem with obvious ground truth and requires a representation and learning algorithm that can capture a high level of musical information. Artist ID problems are usually formed as multi-class problems with a high number of output classes; as a result they benefit from dimensionality reduction steps that reduce noise in the input space.

We start with the set of 51 artists that were used in [1], chosen randomly from a common music testbed which covers most popular genres. We split the set into two subsets: 25 artists for the basis extraction and 26 artists for the artist ID task. Each artist was represented by five songs worth of material chosen randomly (across albums if available.) We compute a feature vector space on the entire music set: after downsampling the audio to 11kHz and removing the mean, for every 5s of audio we compute a 512-point PSD estimate. This left us with roughly 12,000 observations with 257 dimensions each.

Training:	non	pca	nmf	sem	base
Per-class	98.9%	99.3%	95.6%	100%	25%
Per-observation	99.8%	99.9%	99.3%	100%	3.9%

Testing:	non	pca	nmf	sem	base
Per-class	31.6%	28.6%	43.8%	66.3%	25%
Per-observation	22.2%	24.6%	19.5%	67.1%	3.9%

Table 2: Training and testing results for artist ID experiment. Per-class accuracy is the $P(t_p)P(t_n)$ measure for RLSC bias correction averaged over all class t. Per-observation accuracy is a more natural metric: for each observation, was the artist classifier correct?

4.2. Computing the Reductions

We choose $r = 10$, and compute the PCA on the basis extraction set. We store only the transform weight matrix $\mathbf{PCA_w}$. We also compute the NMF in the same manner (over 5,000 iterations) and store its $\mathbf{NMF_w}$. We then compute the semantic classifiers. We subdivide the 25 artist set into two sets of artists (train = 12 artists, test = 13) since labels are tied at the artist level, and use Klepmit to get the set of $P(t_a)$ as in Section 3. After performing the RLSC step (with $\mathbf{C} = 10$ and $\sigma = 0.5$) we evaluate against the 13-artist set and retain the top 10 from our sorted list of $\mathbf{c_t}$ classifiers.

4.3. Applying the Reductions

We then apply the stored $\mathbf{PCA_w}$ and $\mathbf{NMF_w}$ to the new 26-artist set used for the artist ID task. Each process creates an observation matrix of $r = 10$. To obtain the semantic reduction, we evaluate each point in our 26-artist set against the stored $\mathbf{c_t}$, returning 10 scalar values for each classifier. (Note that we do not need to label the artist ID dataset with description after learning the decompositions on other data.) We arrange these results in a row and treat the results as a $r = 10$ observation matrix.

4.4. Artist ID Task

We now use our rank-reduced observations to evaluate a 1-in-26 artist identification task. We use RLSC again as our classifier: we create 26 $\mathbf{c_t}$ machines, one for each artist, and assign a binary truth vector $\mathbf{y_t}$ for each artist with 1 signifying the example frame belongs to artist t and 0 otherwise. We split the 26 artist set into equal sized test and train sets, each set containing half of each artist's songs. We perform four experiments: **(1)** first with no rank-reduction (**non**), each observation is left at $r = 257$. **(2)** We next try the PCA-reduced observations, $r = 10$ (**pca**). **(3)** We then use the NMF-reduced observations, $r = 10$ (**nmf**). **(4)** Lastly we use the semantic decomposition described above with $r = 10$ (**sem**).

The results for each experiment are shown in Table 2 along with the baseline (random) results. Confusion matrices for each experiment are in Figure 3. We see overall very high accuracy in

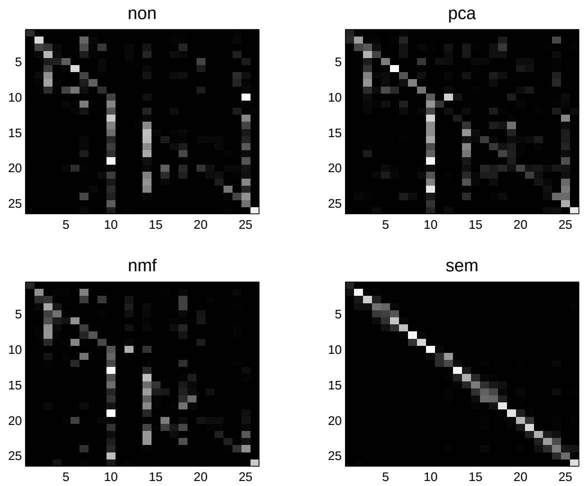

Figure 3: Confusion matrices for the four experiments. Top: no dimensionality reduction and PCA with $r = 10$. Bottom: NMF with $r = 10$ and semantic rank reduction with $r = 10$. Lighter points indicate that the examples from artists on the x-axis were thought to be by artists on the y-axis.

training across the board, with perhaps the NMF hurting the accuracy versus not having an reduced rank representation at all. For the test case, results widely vary. PCA shows a slight edge over no reduction in the per-observation metric while NMF appears to hurt accuracy. We believe the NMF step is not a good fit for noisy audio observations where data is specifically not harmonic and easily separable. However, the semantic rank reduction step appears to do a good job in clustering the observations into a low dimensionality. It far exceeds the accuracy of a PCA pre-processing step and proves to be better than not doing any rank-reduction at all. Clearly the semantic reduction is ‘forcing’ the artist classifier to consider meaningful spectral characteristics not obviously present from statistical analyses.

5. CONCLUSIONS

We show that paying attention to semantic content allows music understanding systems to better grasp noisy acoustic input. Through a completely autonomous and unsupervised method we provide a way to capture the maximal semantic attachment to incoming audio perception and rank reduce observations while maintaining most of the important musical information.

Our next steps involve a better audio representation to learn against the description vectors, better text crawling and noise reduction methods, and other tests of music understanding tasks using this reduced representation. We are interested in obtaining the optimal semantic reduction classifiers with a better term model and machine listening representation over a very large set of music.

6. ACKNOWLEDGMENTS

We thank Adam Berenzweig, Dan Ellis, Ben Recht and Youngmoo Kim for their input, as well as our anonymous reviewers for their helpful comments.

7. REFERENCES

[1] 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. [Online]. Available: http://web.media.mit.edu/~bwhitman/whitman02musical.pdf

[2] B. Whitman, D. Roy, and B. Vercoe, “Learning a lexicon and lexical relations from machine perception of music,” in *HLT-NAACL Workshop on Learning Word Meanings from Non-Linguistic Data*, 2003. [Online]. Available: http://web.media.mit.edu/ bwhitman/whitman03learning.pdf

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

[4] A. Berenzweig, D. P. Ellis, and S. Lawrence, “Anchor models for similarity and artist classification of music,” in *To appear*, 2003.

[5] C. V. L. G.H. Golub, *Matrix Computations*. Johns Hopkins University Press, 1993.

[6] D. Lee and H. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, no. 401, pp. 788–791, 1999.

[7] 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. [Online]. Available: http://web.media.mit.edu/~bwhitman/whitman02inferring.pdf

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

[9] 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.

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

[11] T. Evgeniou, M. Pontil, and T. Poggio, “Regularization networks and support vector machines,” *Advanced In Computational Mathematics*, vol. 13, no. 1, pp. 1–50, 2000.

[12] 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.

[13] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American Mathematical Society*, vol. 68, pp. 337–404, 1950.

[14] B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio, “Categorization by learning and combining object parts.” [Online]. Available: citeseer.nj.nec.com/heisele01categorization.html

[15] 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.

[16] A. Berenzweig, D. P. Ellis, and S. Lawrence, “Using voice segments to improve artist classification of music,” in *AES-22 Intl. Conf. on Virt., Synth., and Ent. Audio.*, 2002.