Musical Query-by-Description as a Multiclass Learning Problem

Brian Whitman
MIT Media Lab
Music, Mind and Machine Group
20 Ames St E15-491
Cambridge, MA 02139
Phone: (617) 253-0112
Email: bwhitman@media.mit.edu

Ryan Rifkin
MIT Artificial Intelligence Laboratory
Center for Biological and Computational Learning
Cambridge, MA 02139
Phone: (617) 225-9392
Email: rif@alum.mit.edu

***Abstract*— We present the query-by-description (QBD) component of "Kandem," a time-aware music retrieval system. The QBD system we describe learns a relation between descriptive text concerning a musical artist and their actual acoustic output, making such queries as "Play me something loud with an electronic beat" possible by merely analyzing the audio content of a database. We show a novel machine learning technique based on Regularized Least-Squares Classification (RLSC) that can quickly and efficiently learn the non-linear relation between descriptive language and audio features by treating the problem as a large number of possible output classes linked to the same set of input features. We show how the RLSC training can easily eliminate irrelevant labels.**

I. Introduction

Retrieval of digital music is suddenly a confirmed hot topic: a curious coincidence erupting from the intersection the 'new' Internet and the popular perceptual codec MP3. As the dust settles from the upheaval created by the record industry trying to arm itself against future technologies, it is becoming more important than ever to provide a simple but powerful interface for music search. Science has responded with novel approaches to multimedia search and music search in particular: query-by-humming [7], query-by-example, or query-by-style or genre [17], [21].

The most natural interface for finding what you want is still language. Most of our purchasing and listening experiences come from conversations with friends and reading reviews. If we want to hear something relaxed or quiet, we read synopses of records, or ask our friend who knows *everything*. And now with all of future media being centrally available through some future distribution mechanism, we can offer systems that can hear these sorts of queries and act as an 'intelligent expert'– one that knows which songs are quiet, which are danceable, and which are loud.

However, this solution presents a number of intermediate problems. First, we need to label the data in some automatic way [20], and then we need to figure out which terms could be construed as musical ('electronic', 'romantic') and which are not ('popular,' 'talented.') We also are faced with an inordinately large scaling problem: with millions of songs and hundreds of thousands of possible labelings, how could we possibly computationally figure out any automatic relation?

In this article, we present a **query-by-description** system for music, using the tenets of language processing, information retrieval and machine learning. Our system treats the relation between words and audio content as a 'severe multi-class' learning problem: given audio content with thousands of known labels (descriptions), only some of which are even relevant (reflect anything in the underlying audio data), we train a machine for each label. We discuss a novel machine learning technique using regularized least-squares classification (RLSC) that makes such multi-class problems tractable and show how we can eliminate incorrect classifications and increase query-by-description accuracy.

II. Background

Searching for music from the content is a current area of research that has gained striking results for various problems. Systems can 'understand' music enough to classify it by genre or style [17], [6], [21], and [8] and have shown high accuracies in either the score level (where the notes and structure are pre-encoded) or the audio domain (where you use a perceptual representation and a learning mechanism.) However, the link between musical content and language is not as prominent: [8] shows that certain style-related terms such as 'lyrical' or 'frantic' can be learned from the score level. Their data was generated by performers who were viewing the terms and asked to play in that style: a luxury analyses of prerecorded music cannot have.

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. [2] and [3] outline similar work. In the 'opposite' direction, work undertaken in [16] learns the meaning of objects from listening to users describing their utility over a microphone.

III. Data Collection and Representation

Our system operates on links between audio content and textual description culled automatically from the Internet. Below

we describe the approaches for collecting and representing both textual descriptive data and the audio data to which it should relate.

A. Audio Dataset

We use audio from the NECI Minnowmatch testbed (related work analyzes this database in [19], [4], [21], [13].) The testbed includes on average ten songs from each of 1,000 artists. The artist list was chosen as the most popular artists on OpenNap, a popular peer-to-peer music sharing service, in August of 2001. ([20] describes the peer-to-peer collection system.)

For the purposes of our experiment, we chose five songs each randomly selected from 51 randomly chosen artists among the dataset, for a total of 255 songs. Each song has the average length of a pop song (we removed a few outliers that were over six minutes or under two minutes.)

We encoded our song set into the representation used in [21], a system that performed accurate style identification. Our aim was to represent the general 'aboutness' of the music, not anything specifically occurring in the bitstream.

The audio tracks were decimated to a sampling rate of 11,025Hz, converted to mono, and had their mean removed. We then take a 512-point power spectral density (PSD) estimate of every three seconds of audio. The data then is rank-reduced using principal components analysis (PCA) to twenty dimensions. The end result is a 'frame' for every three seconds of audio consisting of twenty dimensions. We store the relationship of frame number to artist for accuracy and precision checking later on.

B. Text Description Classes

We use descriptive classes generated from the "Klepmit" system outlined in [20]. The Klepmit text-set contains vectors of 'community metadata' (descriptive terms with salience weights from community description) for each artist in the Minnowmatch testbed. The idea behind Klepmit's data is to represent an artist by means of Internet-wide description by using data mining and information retrieval techniques. The community metadata vectors are meant to be 'time-aware' – that is, public's perception of artists and the artist themselves change over time, which is a crucial missed point in most music-IR systems. We repeatedly crawl for this data weekly, and the cumulative vectors are organized into five term types: n1 (unigrams), n2 (bigrams), np (noun phrases), adj (adjectives), and art (artist names.)

The data is collected as follows: we first query popular search engines for artist names, and parse the resultant pages for text found around the artist. We use a part-of-speech tagger [5] and a noun-phrase chunker [14] to extract the term types described above. (The art term type is meant to extract 'related artists' found when a review or description tries to explain a similar artist.) We then compute statistics on the extracted terms: we compute the f_t (frequency of a term relating to an artist) and f_d (frequency of the term occurring overall), and use both together to create a weighted salience metric s for each term t:

np Term	Score	adj Term	Score
beth gibbons	0.1648	cynical	0.2997
trip hop	0.1581	produced	0.1143
dummy	0.1153	smooth	0.0792
goosebumps	0.0756	dark	0.0583
soulful melodies	0.0608	particular	0.0571
rounder records	0.0499	loud	0.0558
dante	0.0499	amazing	0.0457
may 1997	0.0499	vocal	0.0391
sbk	0.0499	unique	0.0362
grace	0.0499	simple	0.0354

TABLE I
TOP 10 TERMS (NP AND ADJ SETS) FOR 'PORTISHEAD.'

$$s(t) = \frac{f_t e^{-(log(f_d)-\mu)^2}}{2\sigma^2} \quad (1)$$

We use a μ of 3 and a σ of 0.9 throughout, which we arrived at from analyzing the distribution of the term types. See Table I for a list of sample extracted terms and their weights.

The Klepmit text-set was used successfully as a 'cultural representation' in [21] to classify music by a fine-grained style label, and also proved to work well as an artist similarity measure in [10]. It is meant to capture information about music that can not be easily represented by content-based retrieval methods, and also to model the important long-term time domain.

Of further note is that the Klepmit community metadata vectors are completely obtained by automatic and unsupervised methods. Throughout this process of data collection and analysis, we never self-label the audio content nor do we input our own biases.

IV. LEARNING FORMALIZATION OF QUERY-BY-DESCRIPTION

With the text and audio representations in place, we then move to our model of machine learning to uncover descriptive links from community metadata. Our first step is to treat the system as a classification problem: for each possible descriptive term t, we train a machine $\mathbf{c_t}$ to learn the relation between it and an audio frame. However, the problem has three important caveats that separate it from most classification problems:

- **Surfeit of output classes**: Each audio frame can be related to up to 200,000 terms (in the unconstrained case.) Most artists have community metadata vectors of 10,000 terms at one time. For a standard machine learning technique, this would involve costly multi-class learning and combinations.
- **Classes can be incorrect or unimportant**: Due to the unsupervised and automatic nature of the description classes, many are incorrect (such as when an artist is wrongly described) or unimportant (as in the case of terms such as 'talented' or 'cool'– meaningless to the audio domain.) We would need a system that could quickly fetter out such errant classes.

- **Outputs are mostly negative**: Because the decision space over the entire artist space is so large, most class outputs are negative. In our 51 artist set, for example, only two are described as 'cynical' while 49 are not. This creates a bias problem for most machine learning algorithms and also causes trouble in evaluation.

One possible way to learn this relation is to train a binary classifier on each term type, given the audio frames as input examples. However such training has a large startup time for each new class. We show below a novel algorithm that eliminates this startup time and allows for multiple classes to be tested easily.

A. Handling "Severe Multi-class" Problems With Regularized Least-Squares Classification

Regularized Least-Squares Classification (or regression) is a powerful approach to solving machine learning problems [15]. It is related to the Support Vector Machine [18] 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], [15] has shown that the accuracy of RLSC is essentially identical to that of SVMs.

We begin with a *kernel function* K_f, where $K_f(x_1, x_2)$ is a generalized dot product (in a Reproducing Kernel Hilbert Space [1]) between $\mathbf{x_i}$ and $\mathbf{x_j}$. In our work, we use the Gaussian kernel

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

where σ is a tunable parameter. We form the matrix K, where $K_{ij} \equiv K_f(x_i, x_j)$. Then, training an RLSC system consists of solving the system of linear equations

$$(K + \frac{I}{C})\mathbf{c} = \mathbf{y}, \tag{3}$$

where C is a user-supplied *regularization constant*. The resulting real-valued classification function f is

$$f(x) = \sum_{i=1}^{\ell} c_i K(x, x_i). \tag{4}$$

A key property of this approach is that the solution $\mathbf{c}$ is *linear* in the right-hand side $\mathbf{y}$. We compute and store the inverse matrix $(K + \frac{I}{C})^{-1}$ (this is numerically stable because of the addition of the regularization term $\frac{I}{C}$), then for a new right-hand side $\mathbf{y}$, we can compute the new $\mathbf{c}$ via a simple matrix multiplication.

We are faced with a "severe multi-class" problem, where we have a very large number of different labelings of our data. In our application, a reasonable approach is to consider a labelling to be relevant or useful if we can *learn* the relationship between the input and the labelling (on a held out test set). To this end, we will train a single RLSC classifier for each label under consideration. "Training" each classifier is a matrix multiplication (once we have precomputed $(K + \frac{I}{C})^{-1}$), and, if we store the kernel products between all test and training points, each classifier can also be tested via a single matrix multiplication. The ability to very quickly train powerful classifiers is crucial for this application; in our tests, we use roughly 700 different classes, and training an SVM or neural network for each class would be impractical.

V. Experiments and Results

We evaluated the training system and representation by creating a query-by-description prediction task. This experiment hopes to measure the strength of the connections between music and language by asking the system to label as-yet 'unheard' audio with a description. However, this method has a serious flaw in that we cannot trust our ground truth, and therefore automatically scored results (without human intervention) will be low. So instead we place a high value on the differences between different terms' performance.

For the purposes of the experiment, we used the 51 artist, 255 song set described above and split it into two roughly equivalent-sized sets for training and testing. The test and train data are from different artists and therefore represent different acoustic distributions. For our output classes, we chose to stay only within the adjective term types, and limited the possible number of output classes to those that were used to describe at least two of the artists in our 51-artist space. This left us with roughly 700 output classes, many of which only described a few artists.

We computed the stored kernel outlined above (using a σ of 2 and a C of 10) and proceeded to train a new $\mathbf{c_t}$ for each term t against the training set. $f_t(x)$ for the test set is computed over the each frame x and term t. If the sign of $f_t(x)$ is the same as our supposed 'ground truth' for that artist and t, we consider the prediction successful. The evaluation is then computed on the test set by computing a 'weighted precision': where $P(a_p)$ indicates overall positive accuracy (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 $P(a_p)P(a_n)$.

The computation of each consecutive class takes only seconds, a striking improvement over the standard method of computing 700 SVM classifiers for this problem. The results for a select set of terms are shown in Table II. While the overall accuracy is low, we should consider the extremely low baseline of the problem itself compounded with our low trust in the ground truth used for this evaluation. *We see immediately that more 'musically relevant' terms are predicted with far higher accuracy.* In this manner, we can easily remove low-scoring classes, both for data reduction and for accuracy. This type of evaluation provides keen insights into the amount of descriptive power certain terms have against acoustic content.

We also use these results to visualize the spectral fingerprints of various descriptions. We take the mean of all spectral content

Term	Precision	Term	Precision
acoustic	23.2%	annoying	0.0%
classical	27.4%	dangerous	0.0%
clean	38.9%	gorgeous	0.0%
dark	17.1%	hilarious	0.0%
electronic	11.7%	lyrical	0.0%
female	32.9%	sexy	1.5%
happy	13.8%	troubled	0.0%
romantic	23.1%	typical	0.0%
slow	18.9%	unusual	2.3%
upbeat	21.0%	wicked	0.0%
vocal	18.6%	worldwide	2.8%

TABLE II

SELECTED ADJECTIVE TERMS AND THEIR WEIGHTED PRECISION.

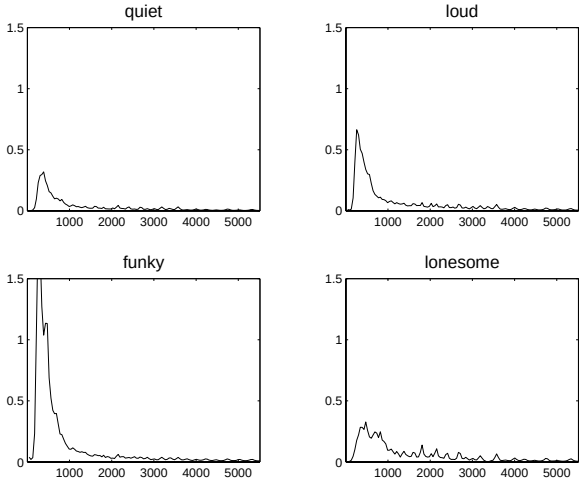

Fig. 1. Mean spectral characteristics of four different terms. Magnitude of frequency on the y-axis, frequency in Hz on the x-axis.

described as certain high-scoring terms, weighting each frame by its gaussian-derived score described above. Figure 1 shows two sets of comparisons. We see the expected result for 'quiet' versus 'loud' and a curious but understandable increase in the bass level bins of the 'funky' spectrum versus 'lonesome"s flat response.

VI. Discussion and Future Work

To achieve a high-accuracy QBD system, we will use the results above to inform a more supervised learning process. By analyzing which terms can be associated with acoustic content, our next step is to use the same algorithms on hand-labeled smaller sets of audio, and then re-label a far larger test set automatically using the thousands of learned relations.

From the results outlined above, we propose that our method of handling a 'severe multi-class' problem such as query-by-description works well at determining incorrect or useless class labels. If we set a threshold ahead of time, the entire process, from data collection to evaluation, could be automated and the set of musically salient terms would be easy to handle.

These experiments clearly show that with enough data and an appropriate statistical measure, we can go a long way towards finding out what we talk about when we talk about music– and gain an overall better understanding of the link between two powerful forms of expression.

VII. Acknowledgements

Thanks to Steve Lawrence and the Music, Mind and Machine group for their helpful contributions.

References

[1] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

[2] K. Barnard, P. Duygulu, and D. A. Forsyth. Clustering art. In *IEEE Conf. on Computer Vision and Pattern Recognition II*, pages 434–441, 2001.

[3] K. Barnard and D. Forsyth. Learning the semantics of words and pictures. 2000.

[4] A. Berenzweig, D. Ellis, and S. Lawrence. Using voice segments to improve artist classification of music. In *AES-22 Intl. Conf. on Virt., Synth., and Ent. Audio*.

[5] E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT, 1992.

[6] W. Chai and B. Vercoe. Folk music classification using hidden markov models. In *Proceedings of International Conference on Artificial Intelligence*, 2001.

[7] W. Chai and B. Vercoe. Melody retrieval on the web. In *Proceedings of Multimedia Computing and Networking*. 2002.

[8] R. B. Dannenberg, B. Thom, and D. Watson. A machine learning approach to musical style recognition. In *In Proceedings of the 1997 International Computer Music Conference*, pages 344–347. International Computer Music Association., 1997.

[9] P. Duygulu, K. Barnard, J. D. Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary.

[10] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The search for ground truth in artist similarity. 2002. To appear.

[11] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advanced In Computational Mathematics*, 13(1):1–50, 2000.

[12] G. Fung and O. L. Mangasarian. Proximal support vector classifiers. In Provost and Srikant, editors, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 77–86. ACM, 2001.

[13] Y. Kim and B. Whitman. Singer identification in popular music recordings using voice coding features. 2002. To appear.

[14] L. Ramshaw and M. Marcus. Text chunking using transformation-based learning. In D. Yarovsky and K. Church, editors, *Proceedings of the Third Workshop on Very Large Corpora*, pages 82–94, Somerset, New Jersey, 1995. Association for Computational Linguistics.

[15] R. M. Rifkin. *Everything Old Is New Again: A Fresh Look at Historical Approaches to Machine Learning*. PhD thesis, Massachusetts Institute of Technology, 2002.

[16] D. Roy. Learning words from sights and sounds: A computational model. 1999.

[17] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals, 2001.

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

[19] B. Whitman, G. Flake, and S. Lawrence. Artist detection in music with minnowmatch. In *Proceedings of the 2001 IEEE Workshop on Neural Networks for Signal Processing*, pages 559–568. Falmouth, Massachusetts, September 10–12 2001.

[20] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proceedings of the 2002 International Computer Music Conference*, pages 591–598. Gothenburg, Sweden, 2002.

[21] B. Whitman and P. Smaragdis. Combining musical and cultural features for intelligent style detection. 2002. To appear.