HIT SONG SCIENCE IS NOT YET A SCIENCE

ABSTRACT

We describe a large-scale and complete experiment aiming at validating the hypothesis that the popularity of music titles can be learnt using global acoustic or human features. We use a 32.000 title database with 632 manually-entered attributes per titles including 3 related to the popularity of the title. We design an experiment with two different audio feature sets, as well as the set of all the manually-entered attributes but the popularity ones. The experiments show clearly that although some subjective attributes may indeed be reasonably well learned by these techniques, it is not the case for popularity. This contradicts recent and sustained claims made in the MIR community as well as in the media about the existence of a so-called "Hit Song Science".

1. INTRODUCTION

The goal of many popular music artists is to create songs that people will like. But music *hits* are as popular as they are mysterious: it is very difficult, in general, to tell why a given song has become, or not, a hit. Nevertheless, recent claims have been made in the community of MIR, as well in the general media, of a possibility to predict if a song will be a hit, using machine-learning techniques. In particular, [4] describe an experiment in which a system is trained to learn a mapping between various musical features extracted from the acoustic signal and from the lyrics, and the popularity (*hitness*) of the song. They conclude from this experiment that their system learns indeed something about popularity, and so that Hit Song Science is indeed possible.

However, the idea that popularity can be inferred from such low-level features contradicts the natural intuitions of any musically-trained listener. Indeed, the experiment described by [4] was performed on a relatively small database (1700 songs). Additionally they used rudimentary features, mostly based on timbre.

In this paper, we describe a larger-scale and more complete experiment designed to further validate this claim. We use a 32.000 song database of popular music titles, associated with fine-grained human metadata, in the spirit of the Pandora effort. To ensure that the experiment is not biased, we use three sets of different features. We describe the various experiments conducted and conclude that popularity is basically not learned by any of these feature sets.

2. EXTRACTING GLOBAL DESCRIPTORS

The most widely used approach to extract global information from acoustic signals is to identify *feature sets*, supposed to be representative of musical information contained in the signal, and to train classifiers such as SVMs on *Training* set, using manually annotated data, aka *ground truth*. These classifiers are then tested, typically on other data sets (the *Test* set), and their performance is evaluated. If the experiment is performed without biases, a good performance of the classifier means that the features considered do carry some information pertaining to the classification problem at hand.

In this paper we describe an experiment similar in sprit to that of [4] on a 32,000 song database. We use three different feature sets to train our classifiers: a *generic* acoustic set *à la* Mpeg-7, a *specific* acoustic set using proprietary algorithms, and a set of high-level metadata produced by humans. These feature sets are described in the next sections.

2.1. Generic Audio Features

The first feature set we consider is related to the so-called *bag-of-frame* (BOF) approach. The BOF approach owns his success to its simplicity and generality, as it can be, and has been, used for virtually all possible global descriptor problems. The BOF approach consists in modelling the audio signal as the statistical distribution of audio features computed on individual, short segments. Technically, the signal is segmented into successive, possibly overlapping frames, from which a feature vector is computed. The features are then aggregated together using various statistical methods, varying from computing the means/variance of the features across all frames to more complex modelling such as Gaussian Mixture Models (GMMs). In a supervised classification context, these aggregated features are used to train a classifier. The BOF approach can be parameterized in many ways: frame length and overlap, choice of features and feature vector dimension, choice of statistical reduction methods (statistical moments or Gaussian Mixture Models), and choice of the classifier (Decision Trees, Support Vector Machines, GMM classifiers, etc.). Many papers in the MIR literature report experiments with variations on BOF parameters on varied audio classification problems [1], [12], [14], [5], [8]. Although perfect results are rarely reported, these works demonstrate that the BOF approach is relevant for extracting a wide range of global music descriptors.

The *generic* feature set we consider here consists of 49 audio features taken mostly from the Mpeg-7 audio standard [7]. This set includes spectral characteristics (Spectral Centroid, Kurtosis and Skewness, HFC, MFCC coefficients), temporal (ZCR, Inter-Quartile-Range), and harmonic (Chroma). These features are intentionally chosen for their generality, i.e. they do not contain specific musical information nor musically *ad hoc* algorithms.
Various experiments (not reported here for space limitations) were performed to yield the optimal BOF parameters for this feature set: localization and duration of the signal, statistical aggregation operators used to reduce dimensionality, frame size and overlap. The best trade-off between accuracy and computation time is achieved with the following parameters: 2048 sample frames with a 50% overlap computed on a 2-minute signal extracted from the middle part of the title, the features are the two first statistical moments of this distribution, i.e. the mean and variance, are considered, yielding a total feature vector of dimension 98 (49 means + 49 variances).

2.2. Specific Audio Features

The *specific* approach consists in training the same (SVM) classifier with a set of "black-box" acoustic features developed especially for popular music analysis tasks by Sony Corporation. These proprietary features have been used in commercial applications such as hard disk based Hi-Fi systems. Altogether, the specific feature set also yields a feature vector of dimension 98, which guaranties a fair comparison with the generic feature set. As opposed to the generic set, the specific set does not use the *BOF* approach: each feature is computed on the whole signal, possibly integrating specific musical information. For instance, one feature describes the proportion of perfect cadences (i.e. resolutions in the main tonality) in the whole title. Another one represents the proportion of percussive sounds to harmonic sounds. We cannot provide here a detailed description of these features as we are mostly interested in comparing the performances of acoustic classifiers on two reasonable, but different feature sets.

2.3. Human Features

Lastly, we trained a classifier with human-generated features. More precisely, we used the attributes provided by our manually annotated database (HiFind, see following section), to infer the popularity attribute. We used 632 Boolean attributes to train the classifiers. This is not directly comparable to the 98 audio features as these attributes are Boolean (and as float values). However, as we will see, these features are good candidate for carrying high-level and precise musical information that are typically not well learnt from audio features.

3. THE HIFIND DATABASE

3.1. A Controlled Categorization Process

Several databases of annotated music have been proposed in the MIR community, such as the RWC database [6], the various databases created for the *Ismir* contests [3]. However, none of them has the scale and number of attributes needed to test our hypothesis. For this study we have used a music and metadata database provided by the HiFind Company (*tm*). This database is a part of an effort to create and maintain a large repository of fine-grained musical metadata to be used in various music distribution systems, such as playlist generation, recommendation, advanced music browsing, etc. The HiFind attributes are binary (0/1 valued) for each song. They are grouped in 16 categories, representing a specific dimension of music: *Style*, *Genre*, *Musical setup*, *Main instruments*, *Variant*, *Dynamics*, *Tempo*, *Era/Epoch*, *Metric*, *Country*, *Situation*, *Mood*, *Character*, *Language*, *Rhythm* and *Popularity*. Attributes describe a large range of musical information: objective information such as the "presence of acoustic guitar", or the "tempo range" of the song, as well as more subjective characteristics such as "style", "character" or "mood" of the song. The *Popularity* category contains three (Boolean) attributes, *low*, *medium* and *high*. It represents the popularity of the title, as observed e.g. from hit charts and records of music history. These three attributes are, in principle, mutually exclusive.
The categorization process at work at HiFind is highly controlled. Each title is listened to entirely by one categorizer. Attributes to be set to *true* are selected using an *ad'hoc* categorization software. Attribute categories are considered in some specific order. Within a category, some rules may apply that prevent some combinations of attributes to be selected. The time taken, for a trained categorizer, to categorize a single title is about 6 minutes. The categorized titles are then considered by a categorization supervisor, who checks, among other things, aspects such as consistency and coherence to ensure that the description ontologies are well-understood and utilized consistently across the categorization team. Although errors and inconsistencies can be made during this process, it nevertheless guaranties a relative good "quality" and consistency of the metadata, as opposed for instance to collaborative tagging approaches in which there is no supervision. Additionally the metadata produced is extremely precise (up to 948 attributes can be considered per title), a precision which is difficult to achieve with collaborative tagging approaches.
There is no systematic way to ensure that the categorization produces absolutely correct and consistent information, so we had to consider the database as it was provided as ground truth. Some minor "clean up" was performed before use, by discarding titles with metadata of obviously of low quality. For instance, we discarded songs

having much less attributes set to "true" than the average (37). Additionally, we kept only those attributes for which we had a significant amount of titles (above 20) with the *true* and *false* values, to build training and testing sets of sufficient size. As a result of this cleanup, the total number of titles considered in this study is 32978, and the number of attributes 632. (Note that those attributes correspond to the 632 human features for the experiment described in Section 2.3) Acoustic signals were given in the form of a *wma* file at 128 kbps. This database was used both for training our classifiers and for testing them, as described in Section 4.1.

3.2. Database Redundancy

The HiFind database is *sparse*: the mean number of attributes set to true per song (occupation factor) is 5.8% (i.e. 37 on a total of 632). Sparseness suggests the dominant role of the true-valued attributes compared to false-valued attributes for a given song. It is also *redundant*. For instance, attributes 'Country Greece' and 'Language Greek' are well correlated. More precisely, assuming the 632 attributes are statistically independent, we can easily compute the entropy of the database [13]. Let D denote the number of attributes ($D = 632$), and $(A_i)_{1 \le i \le D}$ the set of attributes modeled as $0/1$ valued random variables. Let $\Pr(\cdot)$ denote the underlying probability distribution of the random vector A (assumed independent). The entropy of the random vector A is then:

$$S(A) = E(\log \Pr(A)) = \sum_{i=1}^{D} E(\log \Pr(A_i))$$

$$S(A) = \sum_{i=1}^{D} [p_i \log p_i + (1 - p_i) \log(1 - p_i)]$$

If the distribution of the attributes was totally random, the database would have an entropy of 632 and if it consisted of only one value, its entropy would be 0. Applying this formula yields a value of 124 for our database. This high redundancy is a sign of the presence of many inter-attribute dependencies that justifies the deployment of a statistical approach to attribute inference [13]. This redundancy justifies the attempt to infer some attributes from others as explained in Section 2.3.

3.3. Assessing Classifiers

To avoid the problems inherent to the sole use of precision or recall, the traditional approach is to use *F-Measure* to assess the performance of classifiers. For a given attribute, the *recall* R is the proportion of positive examples (i.e. the titles that are *true* for this attribute) that were correctly predicted. The *precision* P is the proportion of the predicted positive examples that were correct. When the proportion of positive examples is high compared to that of negative examples, the precision will usually be artificially very high and the recall very low, regardless of the actual quality of the classifier. The F-measure addresses this issue and is defined as:

$$F = 2 \times {R \times P}/{R + P}$$

However, in our case, we have to cope with a particularly unbalanced 2-class (*True* and *False*) database. So the mean value of the F-measure for each class (*True* and *False*) can still be artificially good. To avoid this bias, we assess the performance of our classifiers with the more demanding *min F-measure*, defined as the minimum value of the F-measure for the positive and negative cases. A *min-F-measure* near 1 for a given attribute really means that the two classes (*True* and *False*) are well predicted.

4. EXPERIMENT

4.1. Experiment Design

We first split the *HiFind* database in two "balanced" parts *Train* and *Test*, so that *Train* contains approximately the same proportion of examples and counter-examples for each attributes as *Test*. We obtained this state by performing repeated random splits until a balanced partition was observed. We trained three classifiers, one for each feature set (*generic*, *specific* and *human*). These classifiers all used a Support Vector Machine (SVM) algorithm with a Radial-Basis Function (RBF) kernel, and were trained and tested using *Train* and *Test*. More precisely, each classifier, for a given attribute, is trained on a maximally "balanced" subset of *Train*, i.e. the largest subset of *Train* with the same number of "True" and "False" titles for this attribute (popularity *Low*, *Medium* and *High*). In practice the size of these individual train databases varies from 20 to 16320. This train database size somehow represents the "grounding" of the corresponding attribute. The classifiers are then tested on the whole *Test* base. Note that the *Test* base is usually not balanced with regards to a particular attribute, which justifies the use of the *min-F-measure* to assess the performance of each classifier.

4.2. Random Oracles

In order to assess precisely the performance of our classifiers, we compare them to random oracles. A random oracle is a classifier that yields a random but systematic answer, solely based on the distribution of examples in the training set. A naive random oracle that would always draw the most represented class could have a non-zero (mean) F-measure, but its *min-F-measure* would be 0, by definition. Therefore, we defined a less naive random oracle for our comparison as follows: given an attribute with p positive examples (and therefore N-p negative ones, with N the size of the sample set), this oracle returns *true* with a probability p/N. By definition, the min-F-measure

of a random oracle only depends on the proportion of positive and negative examples in the test database. Roughly speaking, when using our random oracle, an attribute with balanced positive and negative instances has a *min-F-measure* of approximately 50%, whereas an attribute with 200 positive examples (and therefore around 16,000 negative examples) has a min-F-measure of 2.3%. So the performance of the random oracle is a good indicator of the size of the database, and can therefore be used for comparing classifiers as we will see below.

4.3. Evaluation of the Performance of Acoustic Classifiers

4.3.1. Comparison with random oracles

The comparison of the performance of acoustic classifiers with random oracles shows that the classifiers do indeed learn something about many of the HiFind attributes. More than 450, out of 632, are better learned with the acoustic classifiers than with our random oracle. Table 1 indicates, for each feature set, the distribution of the relative performances of acoustic classifiers with regards to random oracles.

Improvement	Specific	Generic
50	8	0
40	12	15
30	43	20
20	111	79
10	330	360
0	128	158

Table 1 Number of attributes for which an acoustic classifier improves over a random classifier by a certain amount. Column "Improvement" reads as follows: there are 111 attributes for which a specific acoustic classifier outperforms a random classifier by +20 (in min-F-measure).

Table 1 also shows that around 130 to 150 attributes lead to low-performance classifiers, i.e. acoustic classifiers that do not perform significantly better than a random oracle (the last row of the table); approximately half of the attributes lead to classifiers that improve over the performance of a random classifier by less than 10; the rest (top rows) clearly outperform a random oracle, i.e. they are well-modeled by acoustic classifiers.

4.3.2. Distribution of performances for acoustic classifiers

At this point, it is interesting to look at the distribution of the performances of these acoustic classifiers. These performances vary from 0% for both feature sets to 74% for the generic features and 76% for the specific ones. The statistical distribution of the performances is close to a power law distribution, as illustrated by the log-log graph of **Figure 1**.

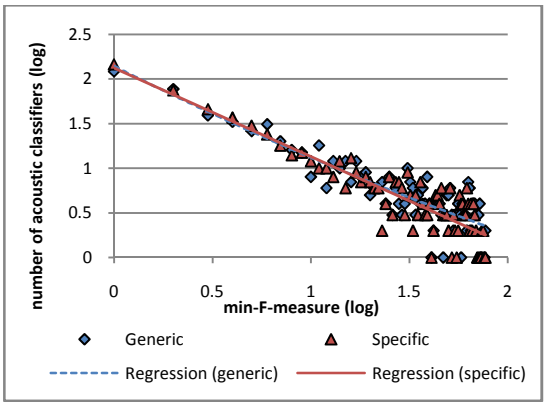

Figure 1. Log-log graph of the distribution of the performance of acoustic classifiers for both feature sets. Triangles (resp. diamonds) correspond to acoustic classifiers trained on generic (resp. specific) features. The dotted (resp. plain) line is a linear regression for the classifiers trained on generic (resp. specific) features. This graph shows that the distribution of the performance of classifiers is close to a power law (with more data fluctuation as we reach high performance, which can be due to the small number of attributes considered, i.e. attributes well-modeled by an acoustic classifier).

These power laws suggest that a natural organization process is taking place in the representation of human musical knowledge, and that the process of automatic audio classification maintains this organization.

4.3.3. Specific features slightly outperform generic features

Not surprisingly, we can see that specific features perform always better than the generic ones. This is illustrated by **Figure 2**. Since the classifiers are both based on the same SVM/kernel, the difference can only come from 1) the actual features extracted or 2) the aggregation method. For the generic features, the aggregation is based on means and averages over all the segments of the song. For the specific features, the aggregation is *ad hoc*.

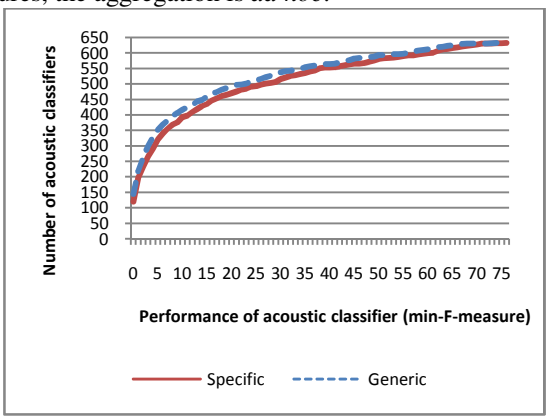

Figure 2. Performance of acoustic classifiers. The dotted and plain lines correspond to generic and specific features respectively. The horizontal axis shows the min-F-measure of the acoustic classifiers. The left-hand (resp.

right-hand) side of the graph corresponds to attributes poorly-modeled (resp. well-modeled) by acoustic classifiers. The dotted line is slightly above the plain line except at the rightmost side. This shows there are more acoustic classifiers with poor performance when using *generic* features, and conversely, more with good performance when using *specific* features.

4.3.4. Acoustic classifiers perform better for large training sets

Lastly, we can observe the relationship between the performance and the size of the training set. The trend lines in **Figure 3** show that the performances of acoustic classifiers increase with the training dataset size, regardless of the feature set. This is consistent with the acknowledged fact that machine-learning algorithms require large numbers of training samples, especially for high-dimensional feature sets.

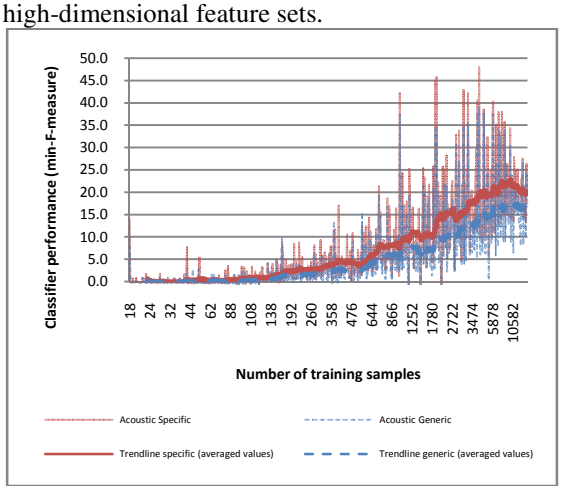

Figure 3. The relative performances of the 632 acoustic classifiers (i.e. the difference between the min-F-measures of the acoustic classifier and of the corresponding random classifier) for specific and generic features, as a function of the training database size. The plain curves correspond to specific features and the dotted curves to generic features. The smooth lines are trend-lines (moving average over 30 values). This graph shows that the performance of the acoustic classifiers increases with the size of the training database.

These experiments show that acoustic classifiers definitely learn some musical information, with varying degrees of performance. It also shows that the subjective nature of the attribute do not seem to influence their capacity to be learned by audio features. For instance, the attribute "Mood nostalgic" is learnt with the performances of 48% (specific features), and 43% (generic features), to be compared to the 6% of the random oracle. Similarly, attribute "Situation evening mood" is learnt with 62% and 56% respectively, against 36% for random. So popularity is, *a priori*, a possible candidate for this task.

4.4. Inference from Human Data

This double feature experiment is complemented by another experiment in which we train a classifier using all the HiFind attributes but the Popularity ones. This is justified by the low entropy of the database as discussed in Section 3.2. Contrarily to the acoustic classifiers, we do not present here the performances of the classifiers for all HiFind attributes. Indeed, some pairs of HiFind attributes are perfectly well correlated so this scheme works perfectly for those, but this result is not necessarily meaningful (e.g. to infer the country from the language). The same *Train / Test* procedure described above applied with the 629 non-popularity attributes as input yields the following result (*min-F-measure*): 41% (*Popularity-Low*), 37% (*Popularity-Medium*) and 3% (*Popularity-High*).

4.5. Summary of Results for Inferring Popularity

The results concerning the Popularity attributes are summarized in Table 2.

Popularity Attribute	Generic Audio Features	Specific Audio Features	Corrected Specific	Human Features	Random Oracle
Low	36	35	31	41	27
Medium	36	34	38	37	22
High	4	3	3	3	3

Table 2 The performances (min-F-measures) of the various classifiers for the three Popularity attributes. No significant improvement on the random oracle is observed.

These results show clearly that the *Popularity* category is not well modelled by acoustic classifiers: its mean performance is ranked fourth on all the (632) attributes considered, but with the second lowest maximum value among categories.

Although these performances appear to be not so bad at least for the "Low" attribute, the comparison with the associated random classifiers shows that popularity is in fact practically not learnt. Incidentally, these performances are not improved with the *correction scheme*, a method that exploits inter-relations between attributes to correct the results [11], in the spirit of the contextual approach described in [2].

Interestingly, the use of human features (all HiFind attributes) does not show either any significant performance.

Lastly, we also considered *a priori* irrelevant information to train our classifiers: the letters of the song title, i.e. a feature vector of size 26, containing the number of occurrences of each letter in the song title. The performances of the corresponding classifiers are respectively 32% 28% and 3% (for the low, medium and

high popularity attributes). This shows that even dumb classifiers can slightly improve the performances of random classifiers (by 5% in this case for the medium and low popularity attributes), but that this information does not teach us anything about the nature of hits.
These results suggest that there are no significant statistical patterns concerning popularity using these features sets.

5. DISCUSSION

We have shown that the notion of music popularity (or *hitness*) of a song cannot be learnt by using state-of-the-art machine learning techniques with two sets of reasonable audio features. This result is confirmed when using supposedly higher-level human metadata. This large-scale evaluation, using the best machine-learning techniques available to our knowledge, contradicts the claims of "Hit Song Science", i.e. that the popularity of a music title can be learned effectively from known features of music titles, either acoustic or human. We think that these claims are either based on spurious data or on biased experiments. This experiment is all the more convincing that some other subjective attributes can indeed be learnt reasonably well using the features sets described here (e.g. the "mood nostalgic" attribute.

This experiment does not mean, however, that popularity cannot be learnt from the analysis of a music signal or from other features. It rather suggests that the features used commonly for music analysis are not informative enough to grasp anything related to such subjective aesthetic judgments. Current works are in progress to determine what are "good" features, in particular works using analytical features [10], which have been shown to outperform manually designed audio features for specific analysis tasks (see e.g. the classification of dog barks, [9]). However, more work remains to be done to understand what are good features of even simpler musical objects such as sounds or monophonic melodies. Hit song science is not yet a science, but a wide open field.

6. REFERENCES

[1] Aucouturier, J.-J. and Pachet, F. (2004) Improving Timbre Similarity: How high is the sky?. Journal of Negative Results in Speech and Audio Sciences, 1(1).

[2] Aucouturier, J.-J., Pachet, F., Roy, P. and Beurivé, A. (2007) Signal + Context = Better Classification. Proceedings of ISMIR 07, Vienna, Austria.

[3] Cano, P. Gómez, E. Gouyon, F. Herrera, P. Koppenberger, M. Ong, B. Serra, X. Streich, S. Wack, N. (2006). *ISMIR 2004 Audio Description Contest*, *MTG Technical Report:* MTG-TR-2006-02.

[4] Dhanaraj, R. and Logan, B. (2005) Automatic Prediction of Hit Songs, Proc. of Ismir 2005, London.

[5] Essid, S. Richard, G. and David, B. (2006) Instrument Recognition in Polyphonic Music Based on Automatic Taxonomies, IEEE Transactions on Speech, Audio and Language Processing, Volume 14, Issue 1, Jan. 2006 Page(s):68 – 80.

[6] Goto, M. Hashigushi, H., Nishimura, T., Oka, R. (2002) "RWC Music Database: Popular, Classical and Jazz Music Databases", Proc. of Ismir 2002, Paris, France.

[7] Kim, H.G. Moreau, N. Sikora, T. (2005), Mpeg7 Audio and Beyond: Audio Content Indexing and Retrieval. Wiley & Sons.

[8] Liu, D. Lu, L. Zhang, H.-J. (2006) Automatic mood detection and tracking of music audio signals, Audio, Speech, and Language Processing, IEEE Trans. on Speech and Audio Processing, 14(1), pp 5-18.

[9] Molnar, C., Kaplan, F., Roy, P., Pachet, F., Pongracz, P., Doka, A. and Miklosi, A. (2008) Classification of dog barks: a machine learning approach, Animal Cognition.

[10] Pachet, F. and Roy, P. (2007) Exploring billions of audio features. In Eurasip, editor, Proc. of CBMI 07.

[11] Pachet, F. and Roy, P. (2008) Improving Multi-Attribute Analysis of Music Titles: A Large-Scale Validation of the Correction Hypothesis, submitted.

[12] Pampalk, E., Flexer, A., Widmer G. (2005) Improvements of Audio-Based Music Similarity and Genre Classification, pp. 628-633, ISMIR 2005.

[13] Rabbat, P. and Pachet, F. (2007) Statistical Inference in Large-Scale Databases: How to Make a Title Rock?, Sony CSL Technical Report CSLP-TR-06-01.

[14] Youngmoo, E. K., Whitman, B. (2002) Singer Identification in Popular Music Recordings Using Voice Coding Features, Proc. of Ismir 2002, pp. 329-336, Paris, France.