

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220723049>

# Automatic Prediction of Hit Songs.

Conference Paper · January 2005

Source: DBLP

---

CITATIONS

88

---

READS

4,133

2 authors, including:



Beth Logan

Institute of Electrical and Electronics Engineers

63 PUBLICATIONS 3,844 CITATIONS

SEE PROFILE



## Automatic Prediction of Hit Songs

Ruth Dhanaraj<sup>1</sup>, Beth Logan  
HP Laboratories Cambridge  
HPL-2005-149  
August 17, 2005\*

Email: [ruthdhan@mit.edu](mailto:ruthdhan@mit.edu), [beth.logan@hp.com](mailto:beth.logan@hp.com)

hit song detection,  
music  
classification

We explore the automatic analysis of music to identify likely hit songs. We extract both acoustic and lyric information from each song and separate hits from non-hits using standard classifiers, specifically Support Vector Machines and boosting classifiers. Our features are based on global sounds learnt in an unsupervised fashion from acoustic data or global topics learnt from a lyrics database. Experiments on a corpus of 1700 songs demonstrate performance that is much better than random. The lyric-based features are slightly more useful than the acoustic features in correctly identifying hit songs. Concatenating the two features does not produce significant improvements. Analysis of the lyric-based features shows that the absence of certain semantic information indicates that a song is more likely to be a hit.

\* Internal Accession Date Only

Published in and presented at the International Conference on Music Information Retrieval, 11-15 September 2005, London, UK

<sup>1</sup> Research Science Institute Intern, HP Labs, Cambridge, MA, USA

Approved for External Publication

© Copyright 2005 Queen Mary, University of London

# 1 Introduction

On April 4 1964 the Beatles accomplished what no band had achieved until then; indeed what no other band has achieved since. In addition to holding the No. 1 USA single with “Can’t Buy Me Love” the Beatles also held the No. 2 slot and the No. 3 slot. In fact Beatles songs occupied the first five positions on the charts. What exactly was it that fueled the Beatles’ rise to fame? Is there an intrinsic quality in music that predisposes it to greatness? In this paper we examine these questions by studying automatic methods to identify hit songs.

The growth of the recording industry has resulted in an abundance of music that requires automated methods of organization and classification. Compression algorithms such as the MP3 file format coupled with connectivity to the internet and improvements in mass storage have contributed to the widespread availability of music in digital form. The size of collective repositories clearly shows a need for organization, but it is also evident that only an automated system is feasible for such massive collections.

Several areas of classification have been proposed and studied. Grouping songs by genre or artist similarity are two common types of classification (e.g. [11, 2]). However seemingly little work has been done on the specific subject of classification and grouping by popularity.

Although societal, cultural, and other qualitative factors undoubtedly play a part in songs’ popularity, in this work we search for some factor that can be quantified in songs which makes them more likely to reach the top of the charts. That is, we assume the group psychology that makes a song popular is not entirely unpredictable, but is somehow based on the qualities of music that appeal to a broad spectrum of people.

Detecting popular songs has tremendous commercial potential; in fact *Hit Song Science*<sup>1</sup> claims to have already succeeded. If properly developed, such technology could help record companies pinpoint the most promising songs and artists and thus better focus their marketing. In this paper we seek to determine if such technology is feasible.

## 2 Methods

There is a large body of literature on song writing and indeed some books even claim to teach how to write a hit song (e.g. [3]). Clearly it isn’t as easy as supposed but certainly melody, chords, lyrics and instrumentation play a role. In this study, we use a very simple approach. We extract very general acoustic and lyric-based features from songs then use standard classifiers to separate hits from non-hits. Our methods are described below.

### 2.1 Acoustic Features

Raw acoustic waveforms are of too high dimension and are too redundant for direct use. Therefore, many features have been proposed to represent the salient properties of songs (e.g. see [11] and references). For hit song classification, we seek an unknown intrinsic universal quality. We therefore extract features from each song describing the main sounds present, where these

---

<sup>1</sup><http://www.hitsongscience.com>

sounds are pre-learned from a corpora of widely varying music. Our method is similar to those which convert songs to vectors according to sounds learnt by supervised clustering (e.g. [5, 1]). We use unsupervised clustering since we don't know which sounds would be optimal for our classification task.

Similar to previous studies, we characterize sounds using MFCC features thus focusing on timbral aspects of the music. Many other characterizations are possible however such as those based on rhythmic or melodic features. We first convert the songs in the training corpora to a set of MFCC features then use K-means clustering to find the N most prominent clusters. We can then convert any song to an N-dimensional vector representation by computing the likelihoods of the sound represented by each cluster occurring in that song.

## **2.2 Lyric-Based Features**

Lyrics are thought to be a large component of what makes a song a hit so we therefore study features based on song lyrics. In this work, we assume that for each song that we can obtain a transcription of the lyrics, most likely from the internet. Future systems may try to extract lyrics directly from audio. However this is beyond state of the art at present.

Text can be analyzed using various methods. Although we originally tried features based on repeating phrases - inspired by song writing guides that claim hits contain catchy, often repeated phrases - we found these did not give good results. Instead we use features which describe the semantic content of each song. Each song is converted to a vector using Probabilistic Latent Semantic Analysis (PLSA) [7]. Similar to the acoustic features described above, each component of the vector represents the likelihood that the song is about a pre-learned topic. The topics are characterized by the words that appear frequently in them and are learned during an automatic training process. We have previously shown that this technique can be effective for determining similarity between songs based on lyrics [8].

## **2.3 Support Vector Machine Classifiers**

The first classifier we use to separate hits from non-hits is a Support Vector Machine (SVM) (e.g. [4]). SVMs are standard classifiers used in many applications. In their simplest implementation they learn a separating 'thick' hyper-plane between two classes which maximizes the 'margin'. This margin roughly corresponds to the distance between the data points residing at the edges of the hyper-plane.

SVMs have several advantages which make them the classifier of choice in many situations. First they do not require any complex tuning of parameters. Second they exhibit a great ability to generalize given a small training corpora. Finally, they are particularly suited to learning in high dimensional spaces.

## **2.4 Boosting Classifiers**

Another popular classification technique is boosting [9, 6]. Boosting combines hundreds or even thousands of "weak learners" in an optimal way. These weak learners could be any classifier

(even SVMs) although for computational reasons they are typically very simple. Each weak learner focuses its attention on those training vectors where the previous weak learners failed. We use a variant of boosting proposed in [10] in which the weak learners are simple linear classifiers on one dimension. This offers the advantage of being less sensitive to spurious features. Components of the feature vector that do not add any advantage are ignored at the expense of more promising components. Additionally, we are able to analyze the relative importance of each feature in a principled way; a simple inspection of the weak learners highlights those features that contribute most to classification.

### 3 Databases

In this section we describe the databases used in our study.

#### 3.1 Ground Truth

Since we are not aware of any publicly available database of hits and non-hits, we use as ground truth data from the Oz Net Music Chart Trivia Page<sup>2</sup>. This site lists all songs which reached the No. 1 ranking in either the United States, the United Kingdom, or Australia since records of No. 1 songs were kept. We use data from January 1956 to April 2004 producing a list of 4439 hit songs. Note that we only consider number No. 1 songs rather than say Top 40 data.

#### 3.2 Acoustic Data

We use an in-house database of approximately 18,500 songs as acoustic data. This data, was obtained by pooling the personal music collections of several members of staff at Hewlett Packard's Cambridge Research Lab. The collection covers many genres ranging from *Reggae* to *Classical* although rock songs form the majority of the collection, totaling around 13,000 songs.

#### 3.3 Lyric Data

Lyrics are much more easily obtained than audio data as several lyrics repositories are freely available on the internet. However many of these are not standardized or in a format conducive to automated retrieval. Additionally, many of these sites are not comprehensive enough for an effective database.

One site with standardized pages is the Astraweb Lyrics Search site<sup>3</sup>. We used data from this site in our experiments. In total we downloaded lyrics for about 500 artists, totaling around 47000 songs, although some songs were repeated on different albums. We stripped all HTML tags, advertising, and excess information to obtain raw lyrics for the songs.

---

<sup>2</sup>[http://www.onmc.iinet.net.au/trivia/hit list.htm](http://www.onmc.iinet.net.au/trivia/hit%20list.htm)

<sup>3</sup><http://lyrics.astraweb.com>

Note that in many cases, the lyrics from this site are not necessarily a verbatim transcript of each song. For example, sometimes repeats of the chorus are simply denoted by “chorus” or not transcribed at all. This may have been why our attempts to construct features based on repeating properties of the lyrics were less than successful since the transcripts did not always faithfully represent the lyrics.

### **3.4 Experimental Database**

For our experiments we considered songs for which we had both lyric and acoustic data. Of the 4000 or so hit songs, we unfortunately only had both acoustic and lyric data available for 91 songs. To complete the experimental database we sampled from the remaining (non-hit) songs for which both acoustics and lyrics were available to make a set of around 1700 songs total. Our aim in restricting the number of non-hits was to avoid a severely unbalanced database.

## **4 Experiments**

To investigate the performance of our proposed hit song classifiers we conduct a series of experiments described below.

### **4.1 Feature Extraction**

We first convert each song in the 1700 song experimental database to acoustic and lyric-based representations.

As described earlier, the first step in converting each song to an acoustic representation is to learn the  $N$  most prominent clusters in a general set of audio. Specifically, we first convert each song in our training set to a series of 20 dimensional MFCC vectors computed from overlapping 25ms windows sampled each 10ms. We discard the 0th (DC) component of each vector then perform K-means clustering to learn the  $N$  most prominent sounds. For computational reasons, we do not learn these clusters using the full 18,500 songs for which we have acoustic data. Instead we sample from this database using around 200 songs to learn the K-means models.

We then convert each song in our experimental database to a  $N$ -dimensional vector as follows. As before, we convert each song to a series of MFCC vectors. For each vector, we score it against each of the  $N$  clusters and increment a counter for the cluster which scores highest. The normalized set of counts forms the  $N$ -dimensional representation for that song.

Similar to the acoustic case, the first step in converting song lyrics to an  $N$ -dimensional representation is to learn a set of topics from a text corpus. We used our set of lyrics as the corpus and after eliminating stop words trained topics according to the algorithm described in [7] using a dictionary of around 91,000 words. We then score the lyrics for each song against these models to produce a normalized vector of counts similar to the acoustic case above.

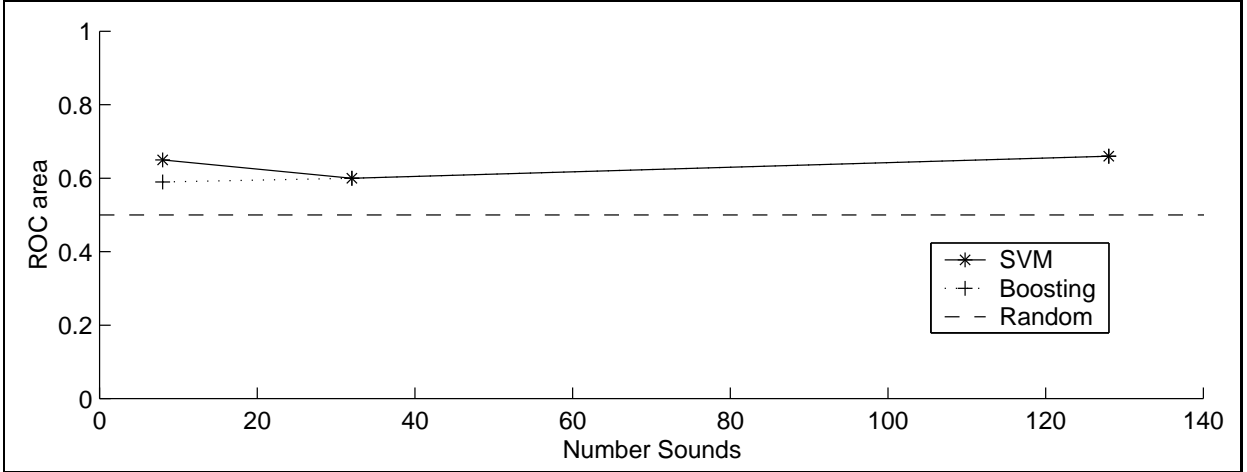


Figure 1: Average ROC area for acoustic-based features with various numbers of sounds for SVM and boosting classifiers

## 4.2 Classification

We run experiments using 10-fold cross validation. This mitigates somewhat the effect of experimenting with such a small database since by averaging over 10 “cuts” of the data into testing and training sets, we reduce the impact of a particularly easy or particularly hard set. Our figure of merit for each classifier is the area under the Receiver Operating Characteristic (ROC) curve. An ROC curve plots sensitivity *vs.* (1-specificity), essentially describing the trade-off between false negatives and false positives as the classifier’s threshold is varied. Random classifiers have ROC area 0.5 and perfect classifiers have ROC area 1.0.

Figures 1 and 2 show the ROC area averaged over the 10 cross validation cuts of the experimental database for SVM and boosting classifiers trained on the acoustic-based and lyrics-based features. We show results for varying numbers of audio sounds and topics. Also shown for reference is the 0.5 ROC area which would result from random classification.

From these plots we see that even with the simple features and classifiers proposed, we can achieve better than random performance. We also see that there the results are comparable for the two classifiers. Lyric-based features appear to give slightly better performance overall than acoustic-based features. The best result obtained using lyrics features is average ROC area 0.68 obtained using 8-topic models. This is slightly better than 0.66, the best result obtained using acoustic features obtained for both 32 and 128-sound models.

We now consider combining acoustic and lyric-based features. We achieve this by concatenating the vectors for the two representations. Figure 3 shows results for this experiment. For simplicity, this plot only shows results for concatenating equal length vectors. For example the “Vector Size 16” result is obtained using feature vectors formed by concatenating length 8 acoustic vectors and length 8 lyric vectors. We also investigated non-equal combinations of acoustic and lyric vectors and saw comparable results.

The best result for combining acoustic and lyric features is average ROC area 0.69. This is obtained by concatenating 32-sound audio features and 8-topic lyric features. This is only

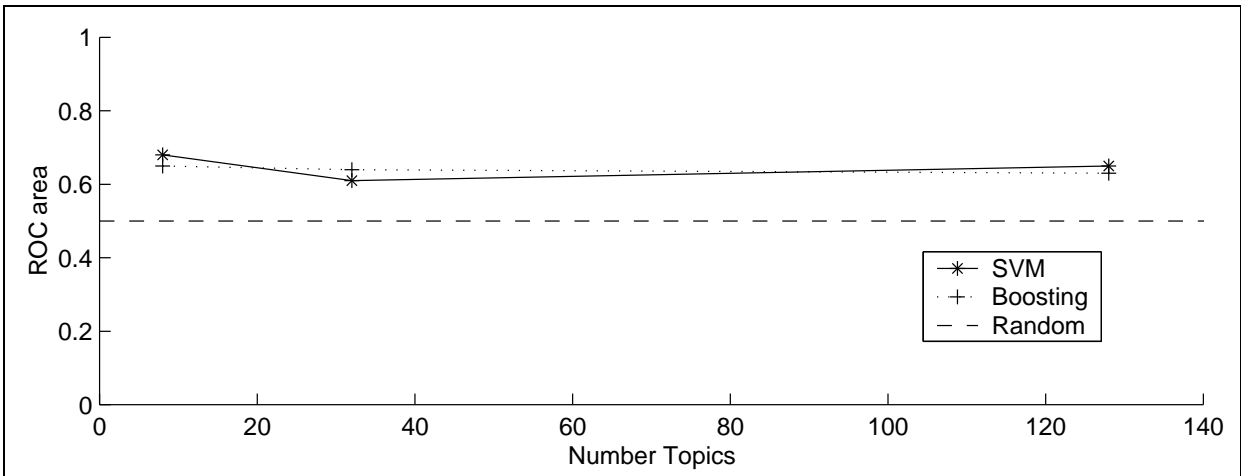


Figure 2: Average ROC area for lyric-based features with various numbers of topics for SVM and boosting classifiers

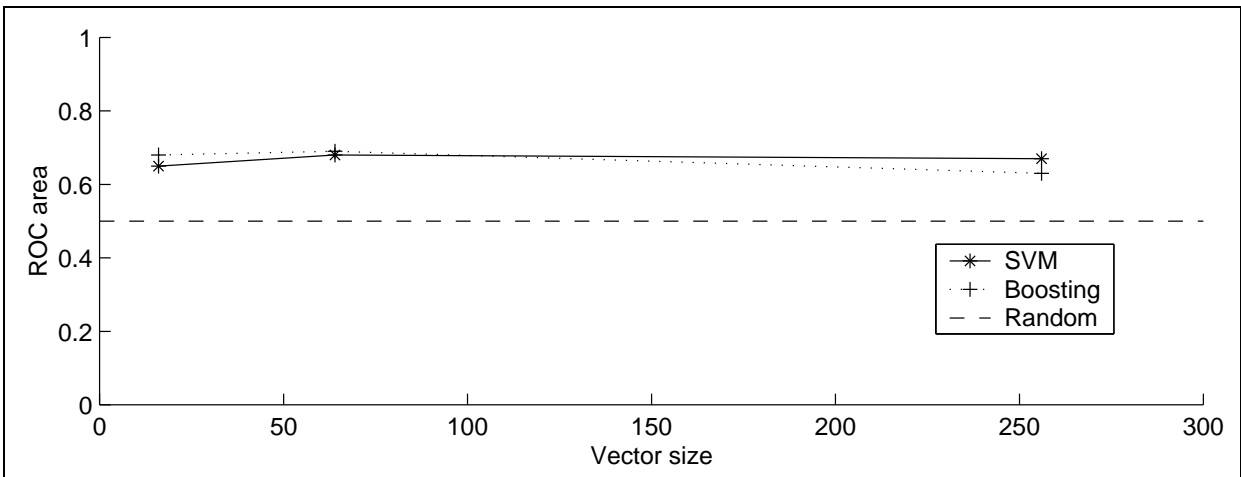


Figure 3: Average ROC area for combined acoustic and lyrics features with varying vector sizes for SVM and boosting classifiers



moderately better than average ROC area 0.68, the best result obtained using lyrics alone. Given the relatively small size of the experiment, it is unclear then whether the combined feature set improves performance.

## 5 Analysis of Results

As discussed earlier, a nice property of the boosting classifier used is that by analyzing the weights of the weak learners we can identify which dimensions of the feature vectors are most helpful for classification. We therefore performed this analysis for the 8-topic lyric vectors.

Table 1 shows the most frequent words which characterize each topic in the 8 topic case. By analyzing the boosting models, we found that by far the most important features for distinguishing hits were Topic 1 and Topic 6. These appear to describe “heavy metal” and “peaceful/new age” music. Interestingly, nearly all of the weak learners learnt *negative* boundaries. That is, the *absence* of Topic 1 or Topic 6 meant the song was more likely to be a hit. Topic 4 which describes generic love songs was the main topic whose presence meant the song was likely to be a hit. However, its impact was much lower than Topics 1 or 6.

Topic	Characterizing Words
0	N*GGA SH*T F*CK YA DONT B*TCH
1	BLOOD CHILDREN WAR DANCE HES
2	DONT YOU'RE SAY THATS MONEY
3	YO DONT CAUSE EM THATS X YALL
4	YEAH OH GIRL HEY SHES BABY
5	LOVE DONT OH YOU'RE BABY SAY
6	AWAY DAY EYES THERES IVE GONE
7	LA QUE DE Y TE BYE MI TU ES YO EN

Table 1: Most frequent words characterizing 8-topic models

## 6 Conclusions and Future Work

Our results suggest that there is indeed some distinguishable thread connecting hit songs. More experimentation is needed, but even our first attempts in this study led to classifiers that are better than random. It seems then that we cannot simply dismiss claims by companies such as *Hit Song Science* as impossible.

Our results indicate that for the features used, lyric-based features are slightly more effective than audio-based features at distinguishing hits. Combining features does not significantly improve performance. Analysis of the best lyric-based system shows that the absence rather than the presence of certain semantic information in the lyrics mean a song is more likely to be a hit. Numerous extensions of this research are possible. For example, future work should examine different weightings of the audio and lyrical data, use larger data sets, and attempt classifica-

tion within smaller groupings, such as music by decade, or music by style. Different kinds of acoustic and lyric features should also be studied. In particular, rhythmic and melodic features merit exploration. Finally, since popular music is continually evolving, time-varying classifiers should be studied.

## References

- [1] A. Berenzweig, D. P. W. Ellis, and S. Lawrence. Anchor space for classification and similarity measurement of music. In *ICME 2003*, pages 29–32, 2003.
- [2] A. Berenzweig, B. Logan, D.P.W. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *Proceedings International Conference on Music Information Retrieval (ISMIR)*, pages 103–109, 2003.
- [3] J. Blume. *6 Steps to Songwriting Success: the Comprehensive Guide to Writing and Marketing Hit Songs*. Billboard Books, revised and expanded edition, 2004.
- [4] C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [5] J. T. Foote. Content-based retrieval of music and audio. In *SPIE*, pages 138–147, 1997.
- [6] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, 1995.
- [7] T. Hofmann. Probabilistic latent semantic analysis. In *Uncertainty in Artificial Intelligence*, 1999.
- [8] B. Logan, A. Kositsky, and P. Moreno. Semantic analysis of song lyrics. In *ICME 2004*, 2004.
- [9] R. E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
- [10] K. Tieu and P. Viola. Boosting image retrieval. In *IEEE International Conference on Computer Vision*, pages 228–235, 2000.
- [11] G. Tzanetakis and P. Cook. Music genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 5(10):293–302, July 2002.