

---

# Linear Programming Boosting for Classification of Musical Genre

---

**Tom Diethé, John Shawe-Taylor**  
Department of Computer Science  
University College London, UK  
{t.diethé, J.Shawe-Taylor}@cs.ucl.ac.uk

## Abstract

Classification of musical genre from raw audio files is a fairly well researched area of music research, and as such provides a good starting point for testing a new algorithm. The Music Information Retrieval Evaluation eXchange (MIREX) is a yearly competition in a wide range of machine learning applications in music. MIREX 2005 included a genre classification task, the winner of which [1] was an application of the multiclass boosting algorithm AdaBoost.MH [2]. It is believed that Linear Programming Boosting (LPBoost) is a more appropriate algorithm for this application due to the higher degree of sparsity in the solutions [3]. The present study aims to improve on the [1] result by using a similar feature set and the multiclass boosting algorithm LPBoost.MC.

## 1 Introduction

A music genre is a subjective categorisation of pieces of music that share a certain style. Any given music genre or sub-genre could be defined by the musical instruments used, techniques, styles, context or structural themes. The groupings of musical genres and sub-genres lead naturally to the idea of a genre hierarchy. However, the distinctions both between individual sub-genres and also between sub-genres and their parent genres are not always clear-cut. While attempts have been made to automatically construct genre hierarchies (e.g. [4, 5]), the performance of such systems does not appear to warrant the additional complexity they entail. As such, the focus of the current research is flat classification.

One of the problems with the grouping of musical pieces into genres is that the process is subjective and is directly influenced by the individual's musical background. This is especially true in sub-genres. Another difficulty is that a single artist or band will often span multiple genres or sub-genres, often within the space of a single album. It becomes virtually impossible to classify the artist or the album into a single genre. The matter is further confused by some genre labels being quite vague and non-descriptive. For example, the genres "World" and "Easy Listening" are often used as a catch-all for music that doesn't naturally fit into more common genres such as rock or classical. However it can be argued that the automatic classification of new material into existing genres is of interest for commercial and marketing reasons.

The performance of humans in classifying musical genre has been investigated in [6]. In this study participants were trained using representative samples from each of the ten genres, and then tested using a ten-way forced-choice paradigm. Participants achieved an accuracy of 53% correct after listening to only 250ms samples and 70% correct after listening to 3s samples. Another study by [7] reports similar results. Although direct comparison of these results with the automatic musical genre classification results of various studies is not possible due to different genre labels and datasets, it is notable that human performance and the automatic retrieval system performance is broadly similar. Moreover, these results indicate the fuzzy nature of musical genre boundaries. It also indicates

the difficulty of gathering ground truth annotations, and explains why some datasets appear to be afflicted with particularly poor annotations.

However, probably the main practical problem for research in the field of automatic music classification is the lack of a freely available high quality dataset. Due to legal obstacles it is not possible to publish datasets of popular music in the way that is possible in other fields, such as text recognition. As a result the datasets that are publicly available consist of “white label” recordings which are ostensibly of poorer quality than mainstream recordings (subjectively in terms of musical quality, but objectively in terms of production quality). The present study uses one publicly available dataset (Magnatune) and one provided by a fellow researcher (Anders Meng, see [7]). The former has been used for the MIREX competition (see below) on more than one occasion, and the latter has been used in studies [8, 7], which will be used for comparison.

There are additional problems that have been noted, such as the “producer effect” or “album effect” [9], where all the songs from a single album share overall spectral characteristics much more than those from other albums from the same artist. This can even extend to greater similarities between artists sharing the same producer than between the artists’ albums.

## 1.1 MIREX

The Music Information Retrieval Evaluation eXchange (MIREX) is part of the annual International Conference on Music Information Retrieval (ISMIR). It takes the form of a series of competitions that have been running since 2004. The 2005 competition included an Audio Genre Classification task, in which the task was classification of polyphonic musical audio into a single high-level genre per example. The audio format for the task was MP3, CD-quality (PCM, 16-bit, 44100 Hz), mono. Full files were used, with segmentation being done at author’s discretion.

Although the categories were organised hierarchically, submitted software was only required to produce classifications of leaf categories. The approach taken at MIREX had the advantage of allowing entrants to treat the problem as either a flat or hierarchical classification problem. In addition all of the recordings used had a unique category label.

Two sets of data were used, ‘Magnatune’<sup>1</sup> and ‘USPOP’<sup>2</sup>. The Magnatune dataset has a hierarchical genre taxonomy, while the USPOP categories are at a single level. The audio sampling rates used were either 44.1 KHz or 22.05 KHz (mono). The results for MIREX 2005 are summarised in table 1 below (see the contest wiki <sup>3</sup> for full results). It should be noted that the statistical validity of the results of the MIREX competitions has recently been called into question [10], due to the testing methods employed. The result is that the reported test accuracies are artificially high, so care must be taken when making direct comparisons.

Table 1: Summary of results of the Audio Genre Classification task from MIREX 2005

Participant	Algorithm	Features	Score
Bergstra et al.	ADABOOST	Aggregated features	82.23%
Mandel & Ellis	SVM	KL-Divergence	78.81%
West	Trees,LDA	Spectral & Rhythmic	75.29%
Lidy & Rauber	SVM	Spectral & Rhythmic	75.27%
Pampalk et al.	1-NN	MFCC	75.14%
Scaringella	SVM	Texture & Rhythmic	73.11%
Ahrendt & Meng	SVM	Auto-Regression	71.55%
Burred	GMM/ML	Aggregated features	62.63%
Soares	GMM	Aggregated features	60.98%
Tzanetakis	LSVM	FFT/MFCC	60.72%

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

<sup>2</sup><http://www.ee.columbia.edu/~dpwe/research/musicsim/uspop2002.html>

<sup>3</sup><http://www.music-ir.org/evaluation/mirex-results/audio-genre/index.html>

## 2 Method

### 2.1 Feature Selection

The various methods for classifying musical genre generally differ in the way that acoustic features are selected, how sub-song level features are aggregated into song-level features, and the machine learning techniques used to classify based on the features. This section describes briefly some different approaches to feature selection, followed by a more detailed examination of the approach taken by [1] that forms the basis of the current study.

The techniques that are employed for extracting acoustic features from musical pieces are inspired by speech perception, signal processing theory, and music theory. In most cases the audio waveform is broken into short frames (in the case of [1] these were 46.44ms in length, or 1024 samples of audio at 22050Hz), and then frame level features are constructed.

### 2.2 Frame level features

#### 2.2.1 Discrete short-term Fourier Transform ( $\mathcal{F}(x)$ )

This is an application of the short-term Fourier Transform on digitised data. Fourier analysis is used to analyse the spectral composition of the frames.

$$z^{STFT}[d, k] = \sum_{n=0}^{N-1} x[n]w[kh_s + f_s - n]e^{-j2\pi dn/f_s} \quad (1)$$

A 512-point transform of each frame was performed, of which the lowest 32 coefficients were retained during experiments.

#### 2.2.2 Real Cepstral Coefficients (RCEPS)

The motivation behind cepstral analysis is the source/filter model used in speech processing. A spectrum can be seen as having two components - a slowly varying part (the filter or spectral envelope) - and a rapidly varying part (the source or harmonic structure). These can be separated by taking a further Fourier Transform of the spectrum. Formally, the real cepstrum of a signal is defined as:

$$\text{real}(\mathcal{F}(x)' \log(|\mathcal{F}(x)|)) \quad (2)$$

where  $\mathcal{F}$  is the Fourier transform and  $\mathcal{F}'$  is the inverse Fourier transform.

#### 2.2.3 Mel Frequency Cepstral Coefficients (MFCC)

This is a measure of the perceived harmonic structure of the sound. It is similar to RCEPS, except that the input  $x$  is first projected according to the Mel-scale. A Mel is a psychoacoustic unit of frequency which relates to human perception, the Mel scale can be approximated from the frequency in  $Hz$  by

$$m = 1127.01048 \log_e(1 + f/700) \quad (3)$$

Where  $f$  is the frequency in  $Hz$ .

#### 2.2.4 Zero Crossing Rate

The Zero Crossing Rate (ZCR) of a signal is the rate of sign changes along the signal. This is a measure which for a single instrument is correlated with dominant frequency. The meaning of this measure is less clear for polyphonic music.

Defining the indicator variable  $v[n]$  as

$$v[n] = \begin{cases} 1, & x[n] \geq 0, \\ 0, & x[n] < 0 \end{cases} \quad (4)$$

and the squared difference  $g[n] = (v[n] - v[n - 1])^2$  then the ZCR over a frame is calculated as

$$z^{ZCR}[k] = \sum_{d=1}^{f_s-1} g[kh_s + f_s - d] \quad (5)$$

The complexity of the ZCR amounts to  $\vartheta(f_s)$  and is the cheapest of the features discussed to extract.

### 2.2.5 Spectral Centroid

The spectral centroid describes the centre of gravity of the octave spaced power spectrum and explains if the spectrum is dominated by low or high frequencies. It is related to the perceptual dimension of timbre. The spectral centroid is formulated as

$$z^{ASC}[k] = \frac{\sum_{d=0}^{f_s/2} \log_2 f_d / 1000 |z^{STFT}[d, k]|^2}{\sum_{d=0}^{f_s/2} |z^{STFT}[d, k]|^2} \quad (6)$$

### 2.2.6 Spectral Spread

The audio spectrum spread describes the second moment of the log-frequency power spectrum. It indicates if the power is concentrated near the centroid, or if it is spread out in the spectrum. A large spread could indicate how noisy the signal is, whereas a small spread could indicate if a signal is dominated by a single tone. The spectral spread is formulated as

$$z^{ASS}[k] = \frac{\sum_{d=0}^{f_s/2} (\log_2 f_d / 1000 - z^{ASC}[k])^2 |z^{STFT}[d, k]|^2}{\sum_{d=0}^{f_s/2} |z^{STFT}[d, k]|^2} \quad (7)$$

### 2.2.7 Spectral Rolloff

Spectral rolloff is defined as the  $a$ -quantile of the total energy in  $|\mathcal{F}_s|$ . In other words, it is the frequency under which a fraction of the total energy is found. Formally, let  $K$  be the highest frequency that can be represented by the signal. Then the spectral rolloff is defined by

$$z^{RO}[k] = \max_y \left\{ y : a > \frac{\sum_{k=0}^y |\mathcal{F}_s^{(k)}|}{\sum_{k=0}^K |\mathcal{F}_s^{(k)}|} \right\} \quad (8)$$

The spectral rolloff was calculated at 16 equally spaced thresholds in the interval  $[0, 1]$ .

### 2.2.8 Autoregression (LPC, LPCE)

The  $k$  linear predictive coefficients (LPC) of a signal  $x$  are defined as:

$$z^{LPC}[k] = \arg \min_a \sum_{t=1}^T (x_t - \sum_{i=1}^k a_i x_{t-i}) \quad (9)$$

$$z^{LPCE}[k] = \min_a \sum_{t=1}^T (x_t - \sum_{i=1}^k a_i x_{t-i}) \quad (10)$$

which is equivalent to an autoregressive compression of spectral envelope. The spectral rolloff can be efficiently computed using Levinson-Durbin recursion.

### 2.3 Feature Aggregation

In order to convert the sub-song level feature sets into a manageable feature set for statistical pattern analysis, some form of aggregation of sub-song level features into a single song-level feature set is required.

There are two possible approaches to dealing with the frame-level features. One option is to first classify directly the frame-level features, and combine the outputs of these classifiers into a song-level classification using a scheme (e.g. [11]). A more popular method is to aggregate the features into a single set of song-level features. This can be done by fitting individual Gaussians to each feature (diagonal covariance e.g. [12]), Gaussian densities with full covariance [13], Gaussian mixture models (e.g. [14]), or through an autoregressive model [15].

Bergstra et al chose to use the popular technique of fitting independent Gaussians with diagonal covariance, and additionally experimented with a series of frame sizes. This is the approach taken in the present study. The resulting full feature vector is created by concatenating the means and variances of 256 RCEPS, 64 MFCC, 32 LPC, 1 LPCE, 32 FFTC, 16 rolloff, and 1 ZCR. This leads to  $402 \times 2 = 804$  parameters for each song.

### 2.4 Boosting for Classification

The term boosting describes any meta-algorithm for performing supervised learning, in which a set of “weak learners” create a single “strong learner”. A weak learner is defined to be a classifier which is only slightly correlated with the true classification (i.e. slightly better than chance). By contrast, a strong learner is strongly correlated with the true classification.

Boosting algorithms are typically iterative, incrementally adding weak learners to a final strong learner. At every iteration, a weak learner learns the training data with respect to a distribution. The weak learner is then added to the final strong learner. This is typically done by weighting the weak learner in some manner, usually related to the accuracy of the weak learner. After the weak learner is added to the ensemble, the data is reweighted such that misclassified examples gain weight and correctly classified examples lose weight. The future weak learners then focus more on the examples that are harder to classify.

AdaBoost is an ensemble method that constructs a classifier iteratively. Without *a-priori* knowledge, small decision trees, or decision stumps (decision trees with two leaves) are often used. In this study, decision stumps were used as weak learners. Multiclass classification can be performed using AdaBoost.MH, which is a multiclass and multilabel version of AdaBoost based on the Hamming Loss.

### 2.5 Linear Programming Boosting

The paper by [3] describes a version of boosting using a linear programme approach, which exactly optimises a generalisation error bound. An efficient algorithm LPBoost mimics a simplex based method known as column generation. This involves formulating the problem as if all possible weak hypotheses had already been generated, with the resulting labels becoming the new feature space of the problem. The task that is solved by boosting is to construct a learning function within the output space that minimises misclassification error and maximises the (soft) margin. The authors prove that for the purposes of classification, minimising the 1-norm soft margin error function is equivalent to optimising a generalisation error bound. LPBoost has the advantages over gradient based methods of convergence in a finite number of iterations to a globally optimal solution, and that the resulting solutions are very sparse.

The paper cites results that demonstrate that LPBoost performs competitively with AdaBoost on a variety of datasets. The authors also demonstrate that the algorithm is computationally tractable. For both small and large datasets, the computation of the weak learners outweighs the linear programme running time, which means that LPBoost iterations are in the same order of magnitude as AdaBoost, though slightly higher.

## 2.6 Linear Programming Boosting: Definition

Taking a 1-norm of the slack variables in the margin maximisation framework and optimising the 1-norm of coefficients leads to a linear programme. This can be solved using a simplex-based column generation approach [3], resulting in the algorithm LPBoost. LPBoost can be proved to converge in a finite number of iterations to a globally optimal solution within the hypothesis space. In the dual form the constraints are the weak learners.

The algorithm proceeds by adding a weak learner, and checking if the linear programme is solved. If not then the weak learner is found that violates the constraints the most. This process is repeated until the linear programme constraints are not violated, which leads to the global optimum solution. Although LPBoost iterations are typically slower than AdaBoost, the algorithm converges much more quickly. The LPBoost algorithm is given below.

---

**Algorithm 1** LPBOOST algorithm

---

Given training examples  $(x_1, y_1), \dots, (x_m, y_m), y_i \in \{+1, -1\}$   
Initialise  $D_0(i) = 1/m$   
**while**  $\sum_i D(i)y_i h_t(x_i) \geq \beta$  **do**  
  Update  $D_{t+1}$ : Solve Linear Programme:  
  argmin  $\beta$ ,  
  s.t.  $\sum_i (D(i)y_i h_k(x_i)) \leq \beta, k = 1 \dots t$   
  where  $1/A < D(i) < 1/B$   
**end while**  
Set  $\alpha$  to Lagrangian multipliers

---

## 2.7 Linear Programming Boosting: Implementation

Many linear programmes are too large to consider all the variables explicitly. Since most of the variables will be zero in the optimal solution, only a subset of variables need to be considered. Column generation generates only variables which have the potential to improve the objective function (i.e. negative reduced cost). The problem being solved is split into two problems, known as the master problem and the subproblem. The master problem is the original problem with only a subset of variables, and the subproblem is a new problem created to identify a new variable. The objective function of the subproblem is the reduced cost of the new variable with respect to the current dual variables.

LPBoost was adapted to the multiclass problem using the approach set out in [3]. Specifically,  $h_j(x_i) = 1$  if example  $x_i$  is correctly classified in the appropriate class by weak hypothesis  $h_j$  and  $-1$  otherwise. The predicted class is chosen as the class which achieves a majority in a vote weighted by the ensemble weights of the final set of weak hypotheses. This is the simplest method of boosting multiclass problems, and perhaps not ideal as it requires  $k$  classifiers to be trained where  $k$  is the number of classes. Further investigation of LP multiclass approaches is needed.

## 3 Experiments

The dataset used in the MIREX 2005 genre classification task is not freely available due to licensing issues. Experiments were run using two datasets: an older Magnatune 2004 dataset which is publicly available and a dataset provided by Anders Meng [7].

The RWC Magnatune database used for the MIREX 2004 Audio description contest is still available (see [16]). Whilst this suffers from many of the problems discussed at the beginning of this chapter, it has the advantage of being released under the slightly more lenient framework of the ‘‘Creative Commons’’. The dataset is split into 6 genres (Classical, Electronic, Jazz & Blues, Metal & Punk, Rock & Pop, and World).

The Meng dataset consists of 11 genres, with 1100 training examples and 220 test examples. The integrity of the dataset has been evaluated by humans (experts and non-experts) at a decision time horizon of 30 seconds [7]. It is interesting to note that human performance on this dataset is only at 57.2% in a 11-way forced choice paradigm. This suggests that either the ground truth annotations are

inaccurate or that the genre labels are not very descriptive. The genres in the dataset are Alternative, Country, Easy Listening, Electronica, Jazz, Latin, Pop/Dance, Rap/Hip-Hop, R&B/Soul, Reggae, Rock. However, the dataset was used with some success in previous studies [17, 15].

During the evaluation of this method, two subsets of this dataset were used. The first contained the 2 genres that had the highest rate of accuracy for human performance (Rap/Hip-Hop and Reggae), and the second contained the 4 genres that had the highest rate of accuracy for human performance (Jazz, Pop/Dance, Rap/Hip-Hop, and Reggae). The reasoning behind this was that if the main problems encountered with this dataset were based on inaccuracies or vagaries of the ground truth labelling, these would be reduced by taking the most consistent results from human evaluation.

## 4 Results

The experimental results are summarised in table 2. The AdaBoost stopping parameter and the LPBoost early stopping criteria were selected using 5-fold cross validation. The results shown are average accuracies, with the numbers in parentheses indicating the number of weak learners chosen by the algorithm. The results are in line with expectations, with LPBoost performing competitively with AdaBoost, while the number of weak learners chosen (and hence the number of iterations of the algorithm) is of an order of magnitude lower. This is a significant result in terms of the sparsity of the solutions. The actual classification rates, however, are seemingly quite poor. This is especially true in the 4-class version of the Anders Meng dataset. Although rates of ca. 40% are well above chance (which in the 4 class case would be 25%), the results are some way away from those reported in the MIREX 2005 competition (see table 1 above). However the best reported results for this dataset are 44% for machine classification and 52% for human classification, which indicates that there are problems with the labelling of this dataset.

Table 2: Summary of experimental results. The numbers in parentheses show the number of weak learners in the final solution

Algorithm	Magnatune 6	Meng 2	Meng 4
AdaBoost	61.3% (10000)	87.5% (10000)	43.3% (5000)
LPBoost	63.5% (585)	87.5% (401)	41.7% (452)

## 5 Conclusions

Many different approaches to genre classification have been taken both in terms of feature selection and in terms of algorithm choice. The MIREX 2005 results indicate that boosting with an aggregated feature set works well. However this shows that in a musical sense, the problem is still poorly understood. The short-term spectral features that are commonly used are really only examining different aspects of the texture of the sound, and not really the long-term temporal dynamics. Some attempts to look at temporal dynamics using autocorrelation/autoregression have been attempted, but currently these methods do not perform as well as methods based on short-term spectral features. Clearly some way of combining these two methods appears to be desirable. The experimental results using a replication of the AdaBoost currently have not been able to reproduce the results of the MIREX 2005 competition, but are competitive with other studies based on the datasets used, and also with that of human performance. This highlights one of the main problems for audio identification tasks, namely the choice of an appropriate dataset. The experimental results demonstrate that LPBoost produces classification results competitive with AdaBoost, but with solutions that are an order of magnitude more sparse. Further work on approaches to multiclass LPBoost is needed. Two possible alternatives include a structured output learning approach (such as [18]), and an augmented objective function where all of the one-vs-rest problems are solved in a single linear programme.

## References

- [1] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and K. Balázs. Aggregate features and AD-BOOST for music classification. *Machine Learning*, 65 (2-3):473–484, 2006.

- [2] R.E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37:297–336, 1999.
- [3] Ayhan Demiriz, Kristin P. Bennett, and John Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1–3):225–254, 2002.
- [4] Juan Jos Burred and Alexander Lerch. A hierarchical approach to automatic musical genre classification. In *Proceedings of the 6th International Conference on Digital Audio Effects*, September 2003.
- [5] Stefan Brecheisen, Hans-Peter Kriegel, Peter Kunath, and Alexey Pryakhin. Hierarchical genre classification for large music collections. In *IEEE 7th International Conference on Multimedia & Expo*, 2006.
- [6] D. Perrot and R. R. Gjerdigen. Scanning the dial: an exploration of factors in the identification of musical style. In *Proceedings of the 1999 Society for Music Perception and Cognition*, 1999.
- [7] A. Meng. *Temporal Feature Integration for Music Organisation*. PhD thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2006. Supervised by Jan Larsen and Lars Kai Hansen, IMM.
- [8] P. Ahrendt, C. Goutte, and J. Larsen. Co-occurrence models in music genre classification. In V. Calhoun, T. Adali, J. Larsen, D. Miller, and S. Douglas, editors, *IEEE International workshop on Machine Learning for Signal Processing*, pages 247–252, Mystic, Connecticut, USA, sep 2005.
- [9] 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, 2001.
- [10] A. Flexer. Statistical evaluation of music information retrieval experiments. 35(2):113–120, 2006.
- [11] Changsheng Xu, C Namunu, Xi Maddage, and Qi Tian Shao. Musical genre classification using support vector machines. In *International Conference of Acoustics, Speech and Signal Processing (ICASSP03)*, 2003.
- [12] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [13] Michael Mandel and Daniel Ellis. Song-level features and support vector machines for music classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005.
- [14] Jean-Julien Aucouturier and Francois Pachet. Music similarity measures: What’s the use? In Ircam, editor, *Proceedings of the 3rd International Symposium on Music Information Retrieval*, October 2002.
- [15] P. Ahrendt and A. Meng. Music genre classification using the multivariate AR feature integration model, aug 2005. Extended Abstract.
- [16] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical and jazz music databases. In *Proceedings of the International Conference on Music Information Retrieval*, page 2878, 2002.
- [17] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *International Conference on Music Information Retrieval*, pages 604–609, sep 2005. Final version : 6 pages instead of original 8 due to poster presentation.
- [18] Sandor Szedmak, John Shawe-Taylor, and Emilio Parado-Hernandez. Learning via linear operators: Maximum margin regression. Technical report, University of Southampton, 2006.