

# AUTOTAGGING MUSIC USING SUPERVISED MACHINE LEARNING

**Douglas Eck**  
Sun Labs  
Sun Microsystems  
Burlington, Mass, USA  
douglas.eck@umontreal.ca

**Thierry Bertin-Mahieux**  
Univ. of Montreal  
Dept. of Comp. Sci.  
Montreal, QC, Canada  
bertinmt@iro.umontreal.ca

**Paul Lamere**  
Sun Labs  
Sun Microsystems  
Burlington, Mass, USA  
paul.lamere@sun.com

## ABSTRACT

Social tags are an important component of “Web2.0” music recommendation websites. In this paper we propose a method for predicting social tags using audio features and supervised learning. These automatically-generated tags (or “autotags”) can furnish information about music that is untagged or poorly tagged. The tags can also serve to smooth the tag space from which similarities and recommendations are made by providing a set of comparable baseline tags for all tracks in a recommender system.

## 1 INTRODUCTION

In this paper we investigate the automatic generation of tags with properties similar to those generated by social taggers. Specifically we introduce a machine learning algorithm that takes as input acoustic features and predicts social tags mined from the web (in our case, Last.fm). The model can then be used to tag new or otherwise untagged music, thus providing a (partial) solution to the cold-start problem. We believe these autotags might also serve to dampen feedback loops which occur when certain songs in a social recommender become over-popular and thus over-tagged.

Recently, there has been increasing interest in *social tagging* [7] including the social tagging of music. Music tagging sites such as Last.fm and QLOUD (www.qloud.com) allow music listeners to apply free-text labels (tags) to songs, albums or artists. In this paper we attempt to predict social tags directly from audio features. For this research, we extracted tags and tag frequencies for more than 50,000 artists from the social music website Last.fm using the Audioscrobbler web service [1]. The majority of tags describe audio content. Genre, mood and instrumentation account for 77% of the tags. However tags can also be ambiguous and/or malicious. Although these issues make working with tags difficult, they are not impossible to overcome. Some strategies to deal with these are described in [6].

When social tags are used as a part of collaborative filtering systems, there is also the problem of social feedback loops: a song can become popular simply because a

few people start recommending it. This was documented in [8], where a number of artificial music markets were created. A more difficult issue is the uneven coverage and sparseness of tags for unknown songs or artists. Since tags are applied by listeners, it is not surprising that popular artists are tagged much more frequently than non-popular artists. Making recommendations for non-popular and new artists (the cold-start problem) is the primary motivation for this area of research. For new music or sparsely tagged music, we predict social tags directly from the audio and apply these automatically generated tags (called *autotags*) in lieu of traditionally applied social tags. By automatically tagging new music in this fashion, we can reduce or eliminate much of the cold-start problem. A longer version of this paper is available [3].

## 2 AN AUTOTAGGING ALGORITHM

We now describe a machine learning model which uses the *meta-learning* algorithm AdaBoost [4] to predict tags from acoustic features. This model is an extension of a previous model [2] which performed well at predicting music attributes from acoustic features: at MIREX 2005 (ISMIR conference, London, 2005) the model won the Genre Prediction Contest and was the 2nd place performer in the Artist Identification Contest. The model has two principal advantages. First it performs automatic feature selection based on a feature’s ability to minimize empirical error. Thus we can use the model to eliminate useless feature sets. Second, it’s performance is linear in the number of inputs. Thus it has the potential to scale well to large datasets. Both of these properties are general to AdaBoost and are not explored further in this short paper. See [4, 9] for more.

**Acoustic feature extraction:** We generated MP3s from a subset of the tagged artists described above. From these MP3s we extracted several popular acoustic features. Do to space limitations, we do not cover feature extraction in depth here. Please see [2] for details. The features used included 20 Mel-Frequency Cepstral Coefficients, 176 auto-correlation coefficients computed for lags spanning from 250msec to 2000msec at 10ms intervals, and 85 spectrogram coefficients sampled by constant-Q (or log-scaled) frequency. We also tried 12 chromagram coefficients but discarded them because (not surprisingly) they contributed

very little to the final result. For those not familiar with these standard acoustic features, please see [5]. The features were extracted with high temporal precision to preserve spectral and timbral information. Following the strategy of [2] coarser “aggregate” features were generated by taking means and standard deviations of high-temporal precision features over longer timescales, here 5sec.

**Tagging as a classification problem:** Intuitively, automatic labeling would be a regression task where a learner would try to predict tag frequencies for artists or songs. However, because tags are sparse (many artist are not tagged at all) this proves to be too difficult using our current Last.fm / Audioscrobbler dataset. Instead we chose to treat the task as a classification one. Specifically, for each tag we try to predict if a particular artist has “none”, “some” or “a lot” of a particular tag relative to other tags. We label training examples as being in one of these three classes based on the relative number of times that tag has been applied. (See our poster or the technical report cited in the introduction for more details).

**Tag prediction with AdaBoost:** Using MultiBoost.MH a booster is trained to predict the tag (“none”, “some”, “a lot”) directly from the aggregate feature values. The value for a song is taken by voting over the predictions for each aggregate feature. Voting can take place in two ways: we can choose segment winners and then select as global winner the class receiving the most segment votes or we can sum the weak learner values over segments and then take the class with the maximum sum.

### 3 EXPERIMENTS

To test our model we used a subset of data from Last.fm / Audioscrobbler. From the full set of tags we selected 13 tags corresponding to popular genres. We selected these particular tags to be relatively easy to analyze (i.e. it’s not clear how to analyze the performance of a predictor of “fun” or “mellow”). For a full list of tags see the technical report cited in the introduction.

**Results:** We compare our results against a baseline computed using the one-versus-all boosted model from [2]. Unlike our current approach, this model assumes that one and only one tag can be applied to a single song. In order to train and test the model we needed to select a winner. We simply chose the most frequent tag. Mean classification error rate over all classes except Classical was 42% (std=2.22) error by segment and 39% (std=2.32) by song. The classical was not counted because only two significant classes “some/all” and “none” could be generated using available data, yielding error of 13%. These results compare favorably to the baseline one-versus-all results of 62% error by segment and 59% error by song.

As an example of model performance see Table 1 where we compare the nearest neighbors for our predicted tags to those for the original tags. Unfortunately there is not enough space to provide more examples. In general it seems that our predicted tags are comparable in quality to the original tags. That is, our tags have some surprising

errors (Marvin Gaye as a near neighbor to the Beatles?) yet so do the original tags (John Williams as a near neighbor to Mozart?).

Near-neighbor artists		
Seed Artist	Last.fm Tags	Our Prediction
Fatboy Slim	The Prodigy Basement Jaxx Apollo 440	Chemical Brothers Apollo 440 Beck
The Beatles	John Lennon The Beach Boys The Doors	Eric Clapton Marvin Gaye The Rolling Stones
Mozart	Bach Beethoven John Williams	Schubert Haydn Brahms

**Table 1.** A comparison of 3 nearest neighbors for Last.fm tags versus our model predictions. Euclidean distance was used.

### 4 REFERENCES

- [1] Audioscrobbler. Web Services described at <http://www.audioscrobbler.net/data/webservices/>.
- [2] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl. Aggregate features and AdaBoost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.
- [3] D. Eck, T. Bertin-Mahieux, and P. Lamere. Autotagging music using supervised machine learning, 2007. Unpublished 6-page version of 2-page ISMIR 2007 publication.
- [4] Y. Freund and R.E. Shapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, 1996.
- [5] B. Gold and N. Morgan. *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. Wiley, Berkeley, California., 2000.
- [6] Marieke Guy and Emma Tonkin. Tidying up tags. *D-Lib Magazine*. online article.
- [7] Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Position paper, tagging, taxonomy, flickr, article, toread. In *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland*, May 2006.
- [8] M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, February 2006.
- [9] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.