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* including the social tagging of music. Music tagging sites such as QLoud (www.qloud.com) and Last.fm (www.last.fm) and allow music listeners to apply free-text labels (tags) to songs, albums or artists.

The real strength of a tagging system is seen when the tags of many users are aggregated. When the tags created by thousands of different listeners are combined, a rich and complex view of the song or artist emerges. Table 1 show the top 20 tags and frequencies of tags applied to the band "The Shins." From these tags and their frequencies we learn much more about "The Shins" than we would from a traditional single genre assignment of "Indie Rock". Additionally, in previous work [3] it was shown that social tags (in this case from the freedb CD track listing service at `www.freedb.org`) can predict canonical music-industry genre with good accuracy. Thus we lose little and gain a lot by moving from genres to tags.

Tag	Freq	Tag	Freq
Indie	2375	Mellow	85
Indie rock	1138	Folk	85
Indie pop	841	Alternative rock	83
Alternative	653	Acoustic	54
Rock	512	Punk	49
Seen Live	298	Chill	45
Pop	231	Singer-songwriter	41
The Shins	190	Garden State	39
Favorites	138	Favorite	37
Emo	113	Electronic	36

Table 1. Top 20 tags applied to *The Shins*

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] for more.

Acoustic feature extraction: We obtained MP3s from a subset of the tagged artists described above. From these MP3s we extracted several popular acoustic features. Due 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 autocorrelation 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 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 5 sec.

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.

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 extracted tags and tag frequencies for more than $50,000$ artists from the social music website Last.fm using the Audioscrobbler web service [1]. 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").

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 2 where we compare the nearest neighbors for our predicted tags to those for the original tags. 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?). Overall, these preliminary results suggest that autotagging helps solve the cold start problem seen in social-tag-based music recommenders.

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

4 CONCLUSIONS

With these preliminary results conclude that a supervised learning approach to autotagging has merit. Our predictions are noisy and lead to sometimes-counterintuitive near artist predictions. However social tags themselves share these properties. There is much future work to do. One next step (already underway) is to learn a much larger number of tags and combine them using a second stage of learning for similarity prediction. A second step is to compare the performance of our boosted model to other approaches such as SVMs and neural networks. Most importantly, the machine-generated autotags need to be tested in a social recommender. It is only in such a context that we can explore whether autotags, when blended with real social tags, will in fact yield improved recommendations.

5 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] J. Bergstra, A. Lacoste, and D. Eck. Predicting genre labels for artists using freedb. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, 2006.

[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.