Autotagging Music using Supervised Machine Learning

Douglas Eck & Thierry Bertin-Mahieux & Paul Lamere

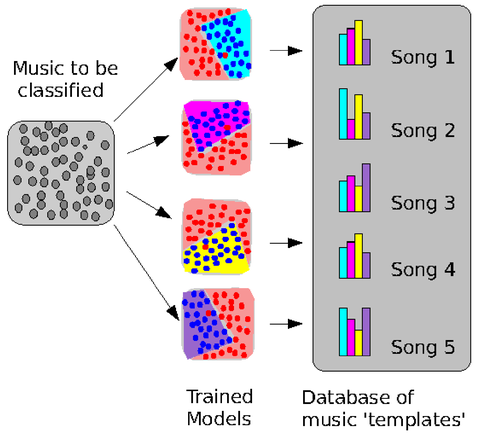
Sun microsystems

Université de Montréal

Overview

Learning to Predict Track-level Social Tags from Audio

- Social tags (STs) plentiful; form usefull similarity space (e.g. last.fm)
- STs suffer from popularity bias, cheating, lack of tags for new music
- Our model predicts STs directly from audio features
- We can then compute similarity using the resulting vector for tracks or artists



We will learn how to tag an unknown song using machine learning. We call these predicted tags **autotags**, and we investigate their merit.

Social Tags

Tag	Freq	Tag	Freq	Tag	Freq
Indie	2375	The Shins	190	Punk	49
Indie rock	1138	Favorites	138	Chill	45
Indie pop	841	Emo	113	Singer-songwriter	41
Alternative	653	Mellow	85	Garden State	39
Rock	512	Folk	85	Favorite	37
Seen Live	298	Alternative rock	83	Electronic	36
Pop	231	Acoustic	54	Love	35

Above are the most frequent tags applied to *The Shins*, below are the types and distribution of the LastFM tags [1].

Type	Freq.	Examples	Type	Freq.	Examples
Genre	68%	heavy metal, punk	Instrumentation	4%	piano, female vocal
Locale	12%	French, Seattle, NYC	Style	3%	political, humor
Mood	5%	chill, party	Misc	3%	Coldplay, composers
Opinion	4%	love, favorite	Personal	1%	seen live, I own it

We use tags applied to artists. We associate them with all songs from that artist. A set of autotags is created for each segment of each song. Artists's autotags are the mean of the segments's autotags.

Machine Learning Algorithm

How can we build a bag of words automatically?
How can we attribute a label to a product based on its attributes?

Audio features

- Classic features used: MFCC, Autocorrelation coefficients, logspec
- Features computed over short time frames, then aggregated [2]

Independance of tags

- Each tag is treated (and learned) independantly
- For one tag and one song, we learn one of the 3 classes y : *no*, *a little* and *a lot*

Algorithm: Classification via adaptive boosting (AdaBoost) of audio features

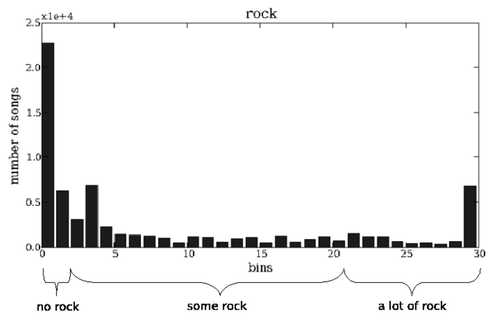
- Simplest classifier: binary decision tree $h(x)$ with 2 leaves, give a value $h_y(x) = -1$ or $+1$ for each class y
- Boosting: a meta-algorithm to combine these classifiers in an iterative way [3]
- Each iteration t, find best classifier $h^t(x)$ on training set, give it a weight α^t based on its performance
- Raise the weights of the wrongly classified examples, therefore give them more importance in the next iteration. Lower the other weights.

$$g_y(x) = \sum_{t=1}^{T} \alpha_y^t h_y^t(x)$$

Usually we obtain a single label by taking the class with the "most votes" i.e $f(x) = \arg\max_y g_y(x)$, but in our model, we use the output value for each class rather than the argmax.

From Tag Count to Classes

example for tag *rock*



We normalize tag count by the maximum tag count for that artist, then we place all songs in an histogram, from which we try to get 3 balanced classes.

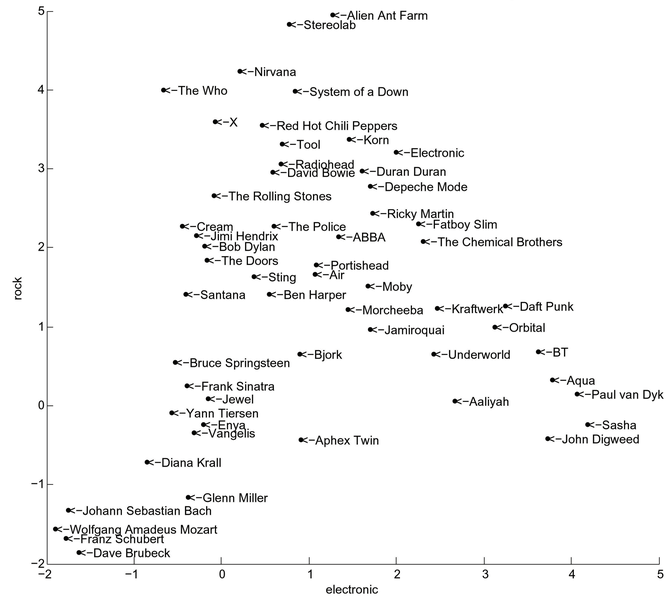
List of Tags

and prediction error per song.

Alternative	40.72 %
Britpop	37.74 %
Classic rock	39.57 %
Classical*	10.00 %
Country	35.15 %
Electronic	38.88 %
Folk	37.36 %
Indie	42.48 %
Jazz	37.53 %
Punk	36.65 %
Reggae	35.18 %
Rock	41.28 %
Soul	41.16 %

* Classical tag has 2 classes

Visualization of Autotags



A set of artists represented by the value of their autotags *rock* and *electronic*. In particular, we see jazz and classical artists get clustered at the bottom left, being neither rock or electronic.

Results per Artist

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

References

[1] Audioscrobbler. Web Services described at http://www.audioscrobbler.net/data/webservices/.

[2] Bergstra, J., Casagrande, N., Erhan, D., Eck, D. and Kegl, B. (2006) Aggregate features and AdaBoost for music classification. *Machine Learning*, 65(2-3):473-484.

[3] Shapire, R.E., and Singer, Y. (1999) Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37(3):297-336.

More references in the paper.

Eck, D., Bertin-Mahieux, T., Lamere, P., Autotagging Music using Supervised Machine Learning, presented at ISMIR 2007, contact: douglas.eck@umontreal.ca