Exploring Automatic Music Annotation with "Acoustically-Objective" Tags

Derek Tingle
Computer Science Department
Swarthmore College
Swarthmore, PA, USA
dtingle1@swarthmore.edu

Youngmoo E. Kim
Electrical & Computer Engineering
Drexel University
Philadelphia, PA, USA
ykim@drexel.edu

Douglas Turnbull
Computer Science Department
Swarthmore College
Swarthmore, PA, USA
turnbull@cs.swarthmore.edu

ABSTRACT

The task of automatically annotating music with text tags (referred to as autotagging) is vital to creating a large-scale semantic music discovery engine. Yet for an autotagging system to be successful, a large and cleanly-annotated data set must exist to train the system. For this reason, we have collected a data set, called *Swat10k*, which consists of 10,870 songs annotated using a vocabulary of 475 *acoustic tags* and 153 *genre tags* from Pandora's Music Genome Project. The acoustic tags are considered "acoustically-objective" because they can be consistently applied to songs by expert musicologists. To develop an autotagging system, we use the Swat10k data set in conjunction with two new sets of content-based audio features obtained using the publicly-available Echo Nest API. The Echo Nest Timbre (ENT) features represent a song using a collection of short-time feature vectors. Compared with Mel-frequency cepstral coefficients (MFCCs), ENTs provide a more compact representation of music and improve autotagging performance. We also evaluate the Echo Nest Song (ENS) feature vector, which is a collection of mid-level acoustic features (e.g., beats per minute, average loudness). While the ENS features generally perform worse than the ENTs, they increase the performance of several individual tags. Furthermore, we plan to publicly release our song annotations and corresponding Echo Nest features so that other researchers will be able to use Swat10K to develop and compare alternative autotagging algorithms.

Keywords

music autotagging, Swat10k, acoustic tags, audio features

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models; H.5.5 [**Sound and Music Computing**]: System

MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.

1. INTRODUCTION

The popularization of digital audio has given music listeners an unprecedented level of access to music [7]. Music discovery sites like Pandora[1] generate novel playlists for individual users based on user preferences. Pandora creates personalized Internet radio stations with songs that are *semantically* similar user-selected artists or songs. Pandora's Music Genome Project employs highly-trained, expert musicologists to semantically analyze music. These musicologists determine the appropriate text-based tags, (e.g., "southern rap influences", "prominent use of the organ", "minor key tonality"), for each song in their corpus. Each musicologist will spend between 20-30 minutes per song selecting the appropriate tags from a vocabulary of several hundred tags. These tags can be considered "acoustically objective" since there is a high-level of agreement between the musicologists when annotating the same song. Once a song has been annotated, it can be compared to other songs based on a comparison of the tags. The quality of these annotations comes at the price of scalability; this method of annotation is extremely human-labor intensive. For this reason, Pandora's song corpus is relatively limited despite having spent many years (and millions of dollars) on music annotation. One potential solution to this problem is content-based autotagging.

Autotagging involves the automatic generation of tags based on an analysis of the audio signal [3, 13, 18, 21]. That is, using human-annotated training songs, we learn a *classifier* that can predict tags for an unannotated song based on audio features that are extracted from the song. The quality and quantity of the training data greatly affects the performance of the autotagging system. One problem facing the Music Information Retrieval (Music-IR) research community is the lack of a large, cleanly-labeled data set. This problem persists partly because of the inability to freely distribute a large corpus of high-quality music without violating copyright law. Another obstacle is the development of a standard vocabulary of music tags [7]. However, the aforementioned cost of training experts and collecting annotations is perhaps the biggest problem for the research community.

To overcome these problems, we have collected a large data set, called *Swat10k*, that consists of 10,870 partially-annotated songs by 4,597 artists. This data set is diverse in that it covers 18 genres ("rock", "jazz", "classical") and 135 subgenres ("indie rock", "bebop", "romantic period opera").

[1] http://www.pandora.com

In addition to genre and subgenre tags, we have mined between 2 and 25 acoustic tags for each song from the Music Genome Project. This acoustic tag vocabulary consists of 475 "acoustically objective" tags which have been provided by trained music experts.

In addition to this data set, we introduce two new sets of audio features. Both sets are obtained using the Echo Nest's public application programming interfaces (API)[2]. While we cannot directly distribute the Swat10k music due to copyright restrictions, all of the audio features presented in this paper are publicly available. More importantly, the Echo Nest offers a host of additional social (e.g, text-mining music blogs, tracking preference information) and acoustic information that we have not explored in this paper. By publishing the Echo Nest song ID for each of the Swat10k songs, other researchers have the ability to extract additional music information using the Echo Nest API.

The first new set of audio features is the set of Echo Nest Timbre (ENT) features. Each song is represented by a bag of short-time audio feature vectors. We show that our state-of-the-art autotagging system is improved when we use ENT feature vectors rather than the commonly used Mel-frequency cepstral coefficients (MFCC) representation. This is a particularly good result because we generally have several hundred ENT feature vectors for the average-length song, whereas we might have tens of thousands of MFCC feature vectors for the same song. This means that the computation required to predict the likelihood of tags is reduced by two orders of magnitude when using ENTs instead of MFCCs.

We also describe a second audio feature representation called the Echo Nest Song (ENS) features. Each song is represented by a single feature vector where each dimension corresponds to a mid-level acoustic characteristic. For example, we use estimates of the beats-per-minute (BPM), the key (e.g., A, A#, B, etc.), the modality (i.e., major or minor), and average loudness. We compare the autotagging performance of two types of discriminative classifiers, Support Vector Machines (SVM) and Boosted Decision Stumps (BDS), trained with song-level Echo Nest features (ENS). Although the short-time features (ENT, MFCC) outperform these song-level (ENS) features over the entire tag vocabulary, we see improvements for several individual tags when using song-level features.

2. RELATED WORK

Early work on content-based audio analysis for text-based music information retrieval focused (and continues to focus) on music classification by genre, emotion, and instrumentation (e.g., [4,11,23]). These classification systems effectively "tag" music with class labels (e.g., "blues", "sad", "guitar"). More recently, *autotagging* systems have been developed to annotate music with a larger, more diverse vocabulary of (non-mutually exclusive) tags [3,13,18,21].

In the 2008 Music Information Retrieval Evaluation eXchange (MIREX), eleven autotagging systems were compared head-to-head in the "Audio Tag Classification" task [1]. Due to multiple evaluation metrics and lack of statistical significance, there was no clear "best" system, though our system was the top performing system for many of the evaluation metrics [21]. Our system uses a generative approach that learns a Gaussian mixture model (GMM) distribution over an MFCC feature space for each tag in the vocabulary. We use this approach for content-based music autotagging when our audio representation is a bag of short-time feature vectors (e.g., MFCC or ENT).

Mandel and Ellis proposed another top performing approach in which they learn a binary SVM for each tag in the vocabulary [13]. They use Platt scaling [16] to convert SVM decision function scores to probabilities so that tag relevance can be compared across multiple SVMs. Eck et. al. [3] also use a discriminative approach by learning a boosted decision stump (BDS) classifier for each tag. We compare the performance of SVMs and BDSs when our audio representation is a single feature vector for each song (e.g., ENS).

Within the Music-IR community, the recent interest in music autotagging has lead to many discussions about creating annotated music data sets. Such data sets are commonly found in other domains such as image annotation (Corel 5k [2], CalTech 101 & 256 [6]) and text classification (Reuters-21578 [10], 20 Newsgroups [8]). To this end, we previously created the CAL500 data set [20]: 500 songs by 500 artists, each of which have been annotated by 3 non-expert undergraduate students using a vocabulary of 174 tags. While this data set has been used by dozens of researchers, it has two major limitations: the small size of the music corpus and the relative subjectivity of many of the tags (e.g., emotions, song usages).

More recently, Law et. al. introduced the MagnaTagatune data set that consists of 6,362 copyright-cleared songs by 269 artists [9]. This data set has been annotated by non-experts using a music annotation called Tagatune[3]. The vocabulary consists of 188 tags that players have used to describe music during the course of game play. For a song-tag annotation pair to be valid, at least two players must independently enter the same tag when listening to the same song. While the music corpus is available for free download and the song annotations are verified by multiple users, there are two main drawbacks of this data set. First, the small number of artists limits the acoustic diversity of the music. Second, and more importantly, the vocabulary is rather simplistic since the game play encourages players to use short, obvious tags such as "singer", "blues", "drums". By comparison, the Swat10k vocabulary includes highly descriptive subgenre tags (e.g., "delta blues") and highly descriptive acoustic tags (e.g., "four-on-the-floor beats").

Perhaps the most common source of training data for content-based autotagging systems is "social" tags that can be harvested from musically-oriented social networks like Last.fm and MyStrands [3, 7, 22]. While this is an abundant source of semantic music information, the data collection process is noisy in that it involves a large community of non-expert and biased fans annotating music with an unconstrained vocabulary of free-text tags. Much like the music annotation game approach, the vocabulary of tags that emerges from social tagging tends to be focused on simplistic and obvious tags. In addition, many tags tend to focus on social, rather than acoustic aspects of the music.

[2]http://developer.echonest.com

[3]http://www.gwap.com/tagatune-o

3. THE SWAT10K ANNOTATED MUSIC CORPUS

The Swat10k data set contains 10,870 songs that are weakly-labeled[4] using a tag vocabulary of 475 *acoustic tags* and 153 *genre tags*. These tags have all been harvested from Pandora's website and result from song annotations performed by expert musicologists involved with the Music Genome Project. We have attempted to collect at least 60 songs from 135 sub-genre radio stations that are produced by Pandora. All of the genres and sub-genres associated with a given song are considered genre tags. For each song in the data set, between 2 and 25 acoustic tags were downloaded from the Pandora music search engine. Because Pandora claims that their musicologists maintain a high level of agreement, we consider the song-tag annotations to be objective. By comparison, other sources of semantic music annotation (social tagging, music annotation games) that are collected from non-experts using an open-ended vocabulary may be more subjective and less oriented towards acoustic characteristics [19].

4. THE ECHO NEST MUSIC API

The Echo Nest is a music technology startup company that was founded in 2005 by Whitman and Jehan, two active Music-IR researchers from the MIT Media Lab. In addition to other services, the Echo Nest provides a free music analysis API that individuals can use to collect information about songs and artists. Once a track has been uploaded and automatically analyzed, the user can query the API using the unique song ID to obtain content-based audio features, as well as socially-oriented information that has been mined from blogs, webpages, social networks, and other sources of music information. Examples of content-based audio features include the estimated times (and associated confidences) of section (e.g., chorus, verse) transitions, bars, beats, and tatums, as well as the predicted key, tempo, time signature, modality and average loudness. In this paper, we focus exclusively on content-based audio features. However, by publishing the Swat10k song list with the associated Echo Nest song IDs, researchers may extract the features that we describe in the following section, as well as additional audio-content and social-context features [22].

5. AUDIO FEATURES

In this section, we introduce two novel audio feature representations, Echo Nest Timbre (ENT) and Echo Nest Song (ENS) features, and compare them to the standard Mel-frequency cepstral coefficients (MFCC) representation [17]. For both the ENT and MFCC representations, each song is represented as a *bag-of-feature-vectors* where each feature vector represents a short-time segment of audio content. These short-time feature vectors can be considered *low-level* because they encode information about the short-term spectral characteristics of the audio track. By contrast, the ENS feature representation is a single feature vector for each song where each dimension corresponds to a specific *mid-level* audio characteristic like the estimated time signature or the estimated modality (i.e., major or minor).

[4]The term "weakly labeled" means that song is labeled with a tag if the tag applies to the song, but the absence of a tag does not necessarily mean that the tag does not apply to the song.

5.1 Mel-frequency Cepstral Coefficients (MFCC)

Mel-frequency Cepstral Coefficients (MFCC) are short-time audio representations that have been used for many speech recognition and music analysis tasks [12, 17]. More specifically, they were incorporated into each of the top performing autotagging systems in the 2008 MIREX tag classification task [1, 3, 13, 21]. An MFCC feature vector is a low-dimensional encoding of the spectral shape of a short-time (23 msec) audio sample. For each song, we select six 5-second intervals that are uniformly spaced throughout the song. Twelve MFCCs are calculated by sliding a half-overlapping 23 msec window over the audio signal. The first and second instantaneous derivatives, referred to as *deltas*, are calculated using the 4 previous and 4 future MFCC feature vectors. We create the 36-dimensional MFCC+Δ by appending the deltas to the MFCC vectors. Note that, for our 30 seconds of audio content, we extract about 2,700 MFCC feature vectors. If we were to use the entire song, we may need to extract tens of thousands of MFCC feature vectors depending on the length of the song.

5.2 Echo Nest Timbre (ENT)

Using the Echo Nest API, we can break a song up into *segments*. Segments are defined as short sound clips (typically between 100-500 msec) with relatively uniform timbre and harmony (e.g., a note). For each audio segment, the Echo Nest calculates 12 "timbre" coefficients that are related to 12 learned basis functions[5]. Each basis function, when represented as a spectrogram in the time-frequency domain, can be loosely related to perceptual qualities such as loudness, brightness, flatness and attack strength.

As with the MFCC representation, we consider two bag-of-feature-vector representations of these timbre-related features: ENT and ENT+Δ. The ENT feature vectors are a time series of 12-dimensional timbre coefficients. Using this time series, we can calculate the first and second instantaneous derivatives for each coefficient to create the 36 dimensional ENT+Δ representation. Unlike the MFCC feature vectors, which are derived from 30 seconds of audio, the ENTs and ENT+Δs are derived from the audio content of the entire song. That is, because segments are typically greater than 100 msec in length, each song can be represented using many fewer ENT feature vectors than MFCC feature vectors. On average, only 785 ENT feature vectors are used to describe each song in the data set compared to 2,736 MFCC feature vectors.

5.3 Echo Nest Song (ENS)

The third set of features are the Echo Nest Song (ENS) features. We compute 34 individual features for each song using music information that is extracted using the Echo Nest API . Ten of these features, including loudness, tempo, key, mode, and time signature, are unmodified Echo Nest features. The remaining features either summarize rhythmic features, (e.g., bars per second, mean bar length, and variance of bar lengths), or relate two rhythmic features, (e.g. ratio of beats to bars.) We have included a full list of the ENS features in the appendix.

[5]The exact calculation of the Echo Nest timbre basis functions is a trade secret of the company.

Table 1: Area under the ROC curve, mean average precision, R-precision, and 10-precision for GMM classifiers trained on short-time features. See text for a description of these evaluation metrics.

		475 Acoustic Tags				153 Genre Tags			
	# Dimensions	AUC	MAP	10-Prec	R-Prec	AUC	MAP	10-Prec	R-Prec
Baseline	-	0.5007	0.0177	0.0148	0.0148	0.4964	0.0126	0.0098	0.0096
MFCC	12	0.8090	0.0898	0.1156	0.1000	0.8557	0.1409	0.1803	0.1531
ENT	12	0.8265	0.1098	0.1435	0.1213	0.8716	0.1731	0.2155	0.1841
MFCC+Δ	36	0.8418	0.1241	0.1622	0.1392	0.8786	0.1957	0.2517	0.2106
ENT+Δ	36	**0.8468**	**0.1317**	**0.1728**	**0.1469**	**0.8874**	**0.2110**	**0.2661**	**0.2244**

6. AUTOTAGGING ALGORITHMS

Autotagging can be considered a multiclass, multilabel classification problem in which each song can be labeled with multiple tags. For each tag in the vocabulary, we train a classifier using the audio features that are extracted from annotated training songs. To annotate a new song, each tag-level classifier is used to predict a relevance score (e.g., probability) that is proportional to the strength of association between the song and the tag. Then during evaluation, we can rank order songs based on their predicted relevance to a given tag.

6.1 Gaussian Mixture Models (GMM)

Our first autotagging algorithm involves learning a Gaussian mixture model (GMM) distribution over an audio feature space (e.g., MFCC, ENT) for each tag in our vocabulary (See [21] for details). First, the expectation-maximization (EM) algorithm is used to learn the parameters of a GMM for each song in our training set. Then, for each tag, we use the Mixture Hierarchies EM algorithm [24] to combine the song-specific GMMs from each positively-labeled training song. The resulting tag-specific GMMs are then used to predict a song-tag likelihood score for each tag in the vocabulary and each song in the test set. These likelihood scores can be interpreted as the parameters of a multinomial distribution over the vocabulary of tags.

6.2 Support Vector Machines (SVM)

In [13], Mandel and Ellis propose using SVM classifiers for autotagging. An SVM is a binary classifier that learns a hyperplane that separates two classes while maximizing the margin surrounding the hyperplane. As in [13], we learn one SVM for each tag using both songs that have been annotated with the tag and songs that have not been annotated with the tag. We then use Platt scaling to output an approximate probability for the tag based on the distance of a data point from the separating hyperplane [16]. In our experiments, we train SVM classifiers using both linear and radial basis function (RBF) kernel functions.

6.3 Boosted Decision Stumps (BDS)

Like SVMs, Boosted Decision Stump (BDS) classifiers are discriminative classifiers that have been used for autotagging music [3]. A *decision stump* is a classifier that focuses on one dimension of a feature vector. If the value of that dimension is above or below a learned threshold, the decision stump outputs a vote in favor of the tag. A BDS classifier is a weighted vote of a set of decision stumps. Using the AdaBoost algorithm [5], we can build up this set by iteratively adding a decision stump and an associated weight. That is, at each iteration of the algorithm, the decision stump that best minimizes the classification error of the training set is selected. The weight is proportional to the reduction in the training set error. As with the SVM classifier, we learn one one-versus-all binary classifier for each tag in the vocabulary. In addition, we can apply Platt scaling to produce probability estimates for a tag given a song.

7. EXPERIMENTS

The first experiment compares the MFCC and ENT feature sets using the GMM classification algorithm. The second experiment evaluates the performance of ENS features when using SVM or BDS classifiers against ENT features when using the GMM algorithm.

In each of our experiments, we train a classifier using 5-fold cross-validation. For cross-validation, we apply an *artist filter* so that all of the songs from each artist appear in either the training set or test set, but not both [15]. This prevents the possibility of overfitting the model to a particular artist in the training set.

The vocabulary of acoustic tags and the vocabulary of genre tags are considered separately. We prune the acoustic tag vocabulary to contain only tags with at least 10 positively-labeled songs in the training set and at least 2 positively-labeled songs in the test set. The tag vocabulary described above is the result of pruning our original vocabulary of over 1,000 unique acoustic tags.

For comparing approaches (defined by a feature set and an autotagging algorithm), we rank order test set songs once for each tag in each vocabulary. For each rank ordering, we use four standard information retrieval metrics to evaluate performance: area under the ROC (AUC), mean average precision (MAP), ten precision (10-Prec), and R-precision (R-Prec). The results are summarized in tables 1 and 2. Each metric in these tables is calculated by averaging the performance of each tag over the five folds, and then averaging over all of the tags in each of our two vocabularies (acoustic tags and genre tags). To calculate the *baseline* scores, each tag rank ordering is a random ordering of the test set songs. The random scores are averaged in the same way as the predicted scores.

The first performance metric is the area under the receiver operating characteristic (ROC) curve (denoted AUC). The ROC curve compares the rate of correct detections to false alarms at each point in the ranking. A perfect ranking (i.e., all the relevant songs at the top) results in an AUC equal to 1.0. We expect the AUC to be 0.5 if we randomly rank songs. Mean average precision (MAP) is found by moving

Table 2: Area under the ROC curve, mean average precision, R-precision, and 10-precision for the BDS classifier (BDS), the SVM classifier with a linear kernel (SVM-L), and the SVM classifier with an RBF kernel (SVM-RBF) trained on song-level features. ENT+Δ is repeated from table 1 for comparison.

	475 Acoustic Tags				153 Genre Tags			
	AUC	MAP	10-Prec	R-Prec	AUC	MAP	10-Prec	R-Prec
Baseline	0.5007	0.0177	0.0148	0.0148	0.4964	0.0126	0.0098	0.0096
ENT+Δ	0.8468	0.1317	0.1728	0.1469	0.8874	0.2110	0.2661	0.2244
SVM-L	0.6182	0.0366	0.0399	0.0399	0.6676	0.0505	0.0637	0.0569
SVM-RBF	0.6913	0.0559	0.0702	0.0648	0.7425	0.0789	0.1036	0.0913
BDS	0.7816	0.0864	0.1507	0.1131	0.8014	0.1139	0.1566	0.1291

down the ranked list of artists and averaging the precision[6] at every point where we correctly identify a relevant song. 10-precision for a tag is the precision when 10 songs are retrieved (e.g., the "search engine metric"). R-precision for a tag is the precision when R songs are retrieved, where R is the number of relevant songs in the ground-truth. More details on these standard IR metrics can be found in Chapter 8 of [14].

When comparing two approaches directly to test whether one is superior to another, we use a one-tailed, paired t-test (with $\alpha = 0.05$) over the individual tag performances (averaged over the five folds, with $n = 475$ or $n = 153$ depending on the vocabulary) on each of the evaluation metrics.

7.1 Short-Time Feature Comparison

Table 1 displays the performance of autotagging using the GMM algorithm trained on four short-time feature representations: MFCC, MFCC+Δ, ENT, ENT+Δ. The left side of the table shows the performance on the 475 acoustic tags and the right side shows the performance on the 153 genre tags. Each choice of feature vectors performs significantly better than random in each of the evaluation metrics. Furthermore, there is a significant improvement when adding the deltas to both MFCCs and ENTs. In comparing MFCC and ENT features of equal dimensionality, the ENT features have significantly higher scores for each metric for both vocabularies.

To summarize, ENT+Δ is the short-time feature that produces the best performance for both acoustic and genre tags. This is an interesting since (a) MFCC is the standard short-time feature representation used by Music-IR community, and (b) ENT represent songs with far fewer feature vectors, and thus provide a more compact representation. This is especially important when considering the computation cost of computing the likelihood of a bag-of-feature-vectors for a large number of songs and a large number of tag-specific GMMs.

7.2 Song-Level Features

Table 2 shows the performance of song-level ENS features when using an SVM classifier with a linear kernel, an SVM classifier with an RBF kernel, and a BDS classifier. The performance of the ENS features using each of the three classifiers is significantly lower than the performance of the GMM classifier trained with ENT+Δ feature vectors. Furthermore, the BDS classifier is significantly better than the SVM-RBF classifier, which is in turn significantly better than the SVM-L classifier across all metrics.

While the ENS features lead to a decrease in performance when averaged across all tags in the vocabulary, on certain tags (e.g. "triple note feel", "a mid-tempo shuffle feel", and "a twelve-eight time signature") ENS features produce large improvements over short-time features. Table 3 presents the five acoustic tags with the greatest performance increase (based on AUC) when using the BDS classifier trained on ENS features compared to the GMM classifier trained on ENT+Δ features. For each of these tags, we show the first five features selected by the BDS classifier.

A qualitative evaluation of the relationship between tags and the most predictive audio features is very interesting. For instance, it is obvious that the *predicted time signature* of a song should improve the classification performance of the tag "a twelve-eight time signature". However, the role of the *variance of bar length* feature with "a twelve-eight time signature" is not as apparent. In music, a twelve-eight time signature means that each measure contains four sets of triplet beats. Human listeners often mistake the time signatures of these pieces to be either 3/4 or 4/4, depending on which subdivision of each beat is stressed. One plausible explanation is that the Echo Nest segmentation algorithm is affected by this same phenomenon, leading to a high variance in the predicted length of each bar.

8. CONCLUSIONS

We present the Swat10k data set which contains 10,780 songs annotated with a large combined vocabulary of 628 tags. In our comparison of short-time audio features, we show that ENT features outperform MFCC features of the same dimensionality using a GMM classifier. In addition to the increased performance, the ENT features require two orders of magnitude fewer feature vectors from each song. We also compare the performance of two SVM classifiers and a BDS classifier when using song-level Echo Nest (ENS) features. These features perform worse than short-time features when averaged over all tags. However, when looking at certain individual tags, autotagging performance is increased when using song-level features.

By making the song & artist names, the Echo Nest song IDs, and the genre & acoustic tags for each song available to the public, we believe that the Swat10k data set will be a useful benchmark data set for the development and evaluation of novel autotagging systems[7].

[6]Precision is the ratio of correctly-labelled songs to the total number of retrieved songs.

[7]To obtain Echo Nest Song IDs for the Swat10k data set, please contact the authors.

Table 3: Acoustic tags that are improved when using Echo Nest Song Features (ENS) and a Boosted Decision Stump classifier (BDS) compared to using Echo Nest Timbre (ENT) Features and a Gaussian Mixture Model (GMM) classifier.

Acoustic tags most improved by BDS classifier		
Tag	Improvement (AUC)	First 5 Features Selected
"triple note feel"	0.1926	time signature, time signature, variance of bar length, ratio of beat length to bar length, tatums per second
"a twelve-eight time signature"	0.1816	key, time signature, variance of bar length, weighted ratio of beat length to bar length, song loudness
"minimalist arrangements"	0.1732	number of sections, song loudness, weighted ratio of tatum length to beat length, mode, end of fade in
"a mid-tempo shuffle feel"	0.0828	song loudness, time signature, start of fade out, weighted ratio of beat length to bar length, variance of tatum length
"use of modal harmony"	0.0782	song loudness, number of sections, variance of section length, number of sections, key confidence

9. REFERENCES

[1] S. J. Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 2008.

[2] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. *European Conference on Computer Vision (ECCV)*, 2002.

[3] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *NIPS*, 2007.

[4] S. Essid, G. Richard, and B. David. Inferring efficient hierarchical taxonomies for music information retrieval tasks: Application to music instruments. *ISMIR*, 2005.

[5] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

[6] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[7] P. Lamere and E. Pampalk. Social tags and music information retrieval. ISMIR Tutorial, 2008.

[8] K. Lang. International conference on machine learning (icml). *Newsweeder: Learning to filter netnews*, 1995.

[9] E. Law and L. von Ahn. Input-agreement: A new mechanism for collecting data using human computation games. In *ACM CHI*, 2009.

[10] D. D. Lewis. Reuters-21578 text categorization test collection. *http://www.daviddlewis.com/resources/testcollections/reuters21578/*, 1997.

[11] T. Li and G. Tzanetakis. Factors in automatic musical genre classification of audio signals. *IEEE WASPAA*, 2003.

[12] Beth Logan. Mel frequency cepstral coefficients for music modeling. *ISMIR*, 2000.

[13] M. Mandel and D. Ellis. Multiple-instance learning for music information retrieval. In *ISMIR*, 2008.

[14] C.D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[15] Elias Pampalk. *Computational Models of Music Similarity and their Application to Music Information Retrieval*. PhD thesis, Vienna University of Technology, 2006.

[16] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 1999.

[17] L. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[18] M. Sordo, C. Lauier, and O. Celma. Annotating music collections: How content-based similarity helps to propagate labels. In *ISMIR*, 2007.

[19] D. Turnbull, L. Barrington, and G. Lanckriet. Five approaches to collecting tags for music. In *ISMIR*, 2008.

[20] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query- by- semantic- description using the CAL500 data set. In *ACM SIGIR*, 2007.

[21] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, 2008.

[22] D. Turnbull, L. Barrington, M. Yazdani, and G. Lanckriet. Combining audio content and social context for semantic music discovery. In *ACM SIGIR*, 2009.

[23] G. Tzanetakis and P. R. Cook. Musical genre classification of audio signals. *IEEE Transaction on Speech and Audio Processing*, 10(5):293–302, 7 2002.

[24] N. Vasconcelos. Image indexing with mixture hierarchies. *IEEE CVPR*, pages 3–10, 2001.

Appendix: Echo Nest Song (ENS) Features

The Echo Nest provides a public API that can be used to call a variety of functions on any uploaded song. To call a function on a specific song, you need the Echo Nest Song ID.

1. song loudness
2. end of fade in
3. start of fade in
4. key
5. key confidence
6. mode
7. tempo
8. tempo confidence
9. time signature
10. time signature confidence
11. number of sections
12. mean section length
13. variance of section length
14. bars per second
15. average bar length
16. variance of bar length
17. weighted average bar length*
18. beats per second
19. average beat length
20. variance of beat length
21. weighted average beat length*
22. tatums per second
23. average tatum length
24. variance of tatum length
25. weighted average tatum length*
26. ratio of avg beat length to avg bar length
27. ratio of avg tatum length to avg beat length
28. ratio of avg tatum length to avg bar length
29. weighted ratio of avg beat length to avg bar length*
30. weighted ratio of avg tatum length to avg beat length*
31. weighted ratio of avg tatum length to avg bar length*
32. ratio of beat count to bar count
33. ratio of tatum count to beat count
34. ratio of tatum count to bar count

* Weighted averages, μ, are calculated by

$$\mu = \frac{\sum_{i=1}^{n} c_i x_i}{\sum_{i=1}^{n} c_i}$$

where x is the predicted length and c is the confidence associated with that prediction.