A Web-Based Game for Collecting Music Metadata

Michael I Mandel
Columbia University
LabROSA, Dept. Electrical Engineering
mim@ee.columbia.edu

Daniel P W Ellis
Columbia University
LabROSA, Dept. Electrical Engineering
dpwe@ee.columbia.edu

August 14, 2008

Abstract

We have designed a web-based game, MajorMiner, that makes collecting descriptions of musical excerpts fun, easy, useful, and objective. Participants describe 10 second clips of songs and score points when their descriptions match those of other participants. The rules were designed to encourage players to be thorough and the clip length was chosen to make judgments objective and specific. To analyze the data, we measured the degree to which binary classifiers could be trained to spot popular tags. We also compared the performance of clip classifiers trained with MajorMiner's tag data to those trained with social tag data from a popular website. On the top 25 tags from each source, MajorMiner's tags were classified correctly 67.2% of the time, while the social tags were classified correctly 62.6% of the time.

1 Introduction

The easiest way for people to find music is by describing it with words. Whether this is from a friend's recommendation, browsing a large catalog to a particular region of interest, or searching for a specific song, verbal descriptions, although imperfect, generally suffice. While there are notable community efforts to verbally describe large corpora of music, e.g. Last.fm, these efforts cannot sufficiently cover new, obscure, or unknown music. Efforts to pay expert listeners to describe music, e.g. Pandora.com, suffer from similar problems and are slow and expensive to scale. It would be useful in these cases to have an automatic music description system, the simplest of which would base its descriptions wholly on the audio and would require human generated descriptions on which to train computer models. An example use of such a system can be seen in Figure 1 along with the data that can be used to train it.

Building a computer system that provides sound-based music descriptors requires human descriptions of the sound itself for training data. Such descriptions are most likely to be applied to *clips*, short segments of relatively obscure songs that provide little context to the listener. Mining available writings on music is not sufficient, as only a small portion of the vast quantity of these, e.g. record reviews and blogs, describes aspects of the sound itself, most describes the music's social context. Broad discussions of genre or style generally focus on the social aspects of music, while specific descriptions of a short segment generally focus on the sound. Similarly, due to the heterogeneous nature of musical style, it is not certain that a description

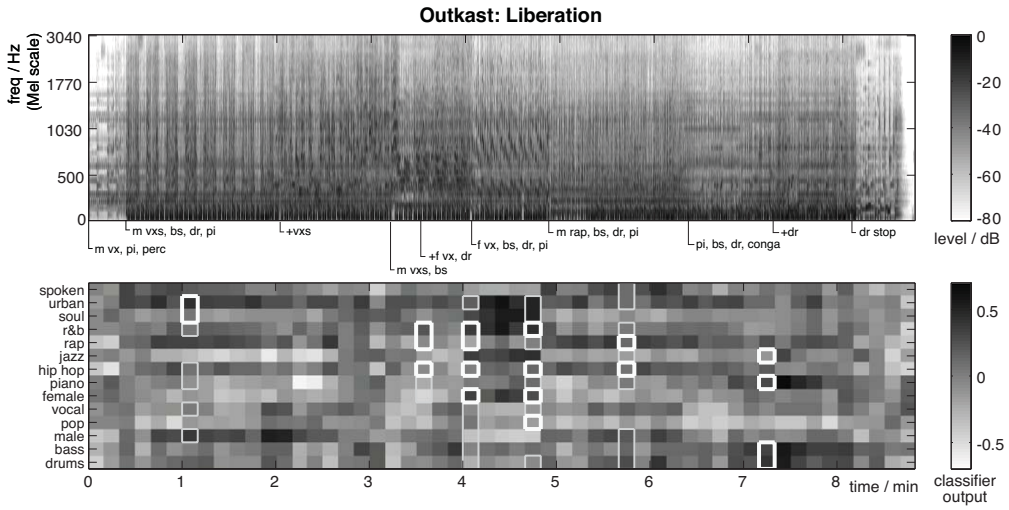


Figure 1: Automatic tagging of all ten second segments within a track, illustrating one goal of this work. The top pane shows the mel-scale spectrogram of OutKast's "Liberation", with major instrumentation changes manually labeled ("m vx" = male voice, "f vx" = female voice, "m vxs" = multiple male voices, "pi" = piano, "bs" = bass guitar, "dr" = drums, "+" indicates instruments added to the existing ensemble). The lower pane shows the output of automatic classifiers trained on the 14 shown labels. Note, for example, how the switch to a single female lead vocal from 4:04 to 4:52 is strongly detected by the labels "female", "jazz", "r&b", "soul", and "urban". The six columns of outlined cells indicate the six clips for which human tags were collected in the MajorMiner game; thin outlines indicate that the tag was used at least once, and thick outlines show the most common tags for each clip. Notice that on the whole human tags are consistent with higher scores from the classifiers.

of a genre or style applies to all possible segments of music within that style. Shorter clips are more likely to be homogeneous, which makes the connection between language and music more definite.

Thus in this project, we endeavor to collect ground truth about specific, objective aspects of musical sounds by asking humans to describe clips in the context of a web-based game[1]. Such a game entertains people while simultaneously collecting useful data. Not only is the data collection process interesting, but the game itself is novel. The overarching goal of collecting thorough descriptions of the music shaped many design decisions about the game-play, including the rules for scoring, the way in which clips are chosen for each player, and the ways in which players can observe each other.

We have also run a number of experiments to test the effectiveness with which an automatic music classifier can be trained on the data collected by such a game, comparing binary classification accuracy for many individual tags on sets of clips that balanced positive and negative examples. We first measured the effect of varying the amount of MajorMiner data available to a classifier, showing that a six-fold increase in training data can improve classification for some tags by ten percentage points. For the other experiments, we used two variations of tags from MajorMiner and three variations of tags from the social music website Last.fm[2]. We compared these datasets on their seven common tags and found that MajorMiner's data were more accurately classified on all but two tags. We also compared classification accuracy for the 25 tags most frequently applied to our music collection by each dataset. Classifiers trained on MajorMiner's data achieve

[1]The game is available to play at http://www.majorminer.com
[2]http://www.last.fm/ and http://www.audioscrobbler.com

Figure 2: A screenshot of a game in progress. The player describes a 10 second clip of an unknown song. Italicized descriptions have scored 1 point, red descriptions 0 points, and gray descriptions have scored no points, but will score 2 points if they are subsequently verified through duplication by another player.

an average accuracy of 67.2% and those trained on Last.fm's data achieve 62.6%.

An example of one possible application of this work is shown in Figure 1. To make this figure, Outkast's song "Liberation" was broken down into contiguous 10-second clips. While a number of these clips were labeled in the MajorMiner game, most of them were not. By training autotaggers on other songs that were tagged in the game, we are able to automatically fill in the descriptions that should be applied to the unseen clips. The agreement between the autotags and the tags collected in the game for this song can be seen in the figure as dark regions of high predicted relevance correspond to outlined regions of human judgements. In addition, the structure of the song becomes visible as the autotags change in response to changes in the music.

1.1 Example of game play

An example session of the game will provide a sense of its rules and strategy. A screenshot of this example session can be seen in Figure 2. First the player, *mim*, requests a new clip to be tagged. This clip could be one that other players have seen before or one that is brand new, he does not know which he will receive. He listens to the clip and describes it with a few words: `slow`[3], `harp`, `female`, `sad`, `love`, `fiddle`, and `violin`. The words `harp` and `love` have already been used by one other player, so each of them scores *mim* one point. In addition, the players who first used each of those words have, at this time, two points added to their scores (regardless of whether they are currently playing the game). Since the words `female` and `violin` have already been used by at least two players, they score *mim* zero points. The words `sad` and `fiddle` have not been used by anyone before, so they score no points immediately, but have the potential to score two points for *mim* at some later time should another player use one.

When the player has finished tagging his clips he can go to his game summary, an example of which can be seen in Figure 3. The summary shows both clips that he has recently seen and those that he has recently

[3]We will use a `monospaced` font to denote tags

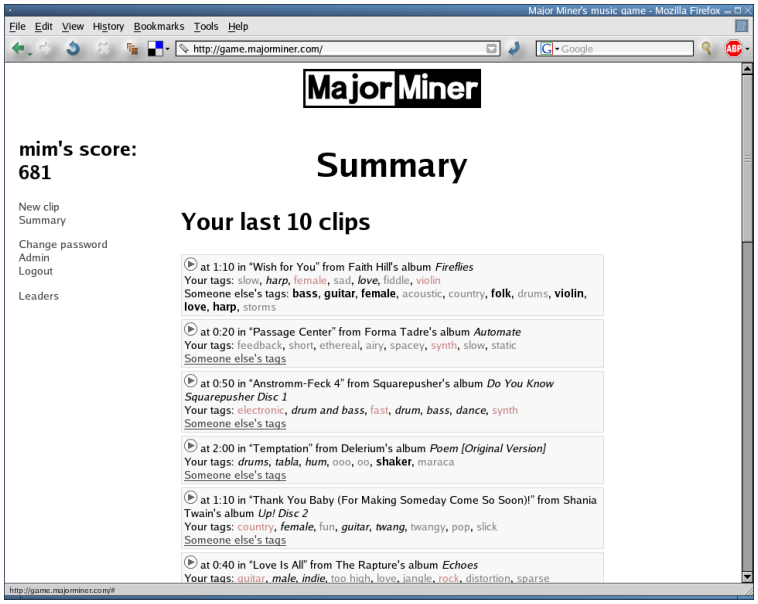


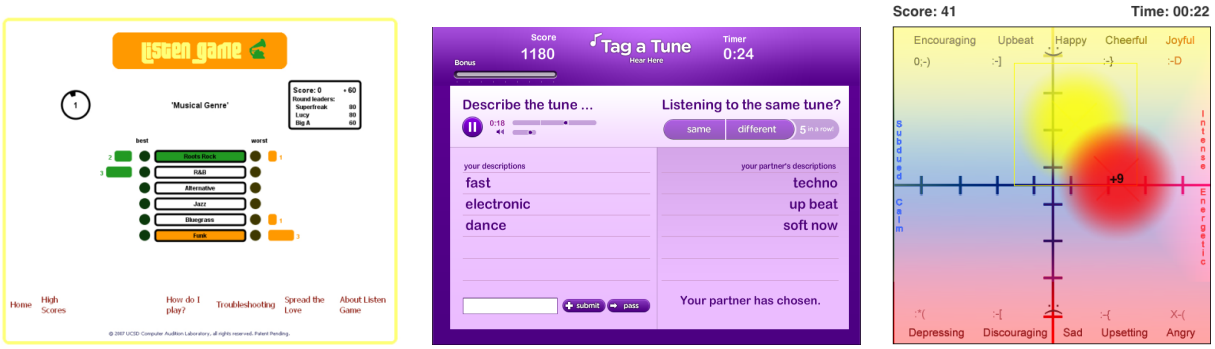
Figure 3: A screenshot of the player's game summary. The artist, album, track, and start time are listed for clips the player has recently seen or scored on. The player can also see their own tags and those of another player.

scored on, e.g. if another player has agreed with one of his tags. It also reveals the artist, album, and track names of each clip and allows *mim* to see one other player's tags for each clip. In the figure, the other player has already scored two points for describing the above clip with `bass`, `guitar`, `female`, `folk`, `violin`, `love`, and `harp`, but has not scored any points yet for `acoustic`, `country`, `drums`, or `storms`. When he is done, *mim* logs out. The next time he logs in, the system informs him that three of his descriptions have been used by other players in the interim, scoring him six points while he was gone.

1.2 Previous work

A number of authors have explored the link between music and text. Whitman and Ellis (2004) trained a system for associating music with noun phrases and adjectives from a collection of reviews from the All Music Guide and Pitchfork Media. This work was based on the earlier work described by Whitman and Rifkin (2002). More recently, Turnbull et al. (2006) used a naive Bayes classifier to both annotate and retrieve music based on an association between the music and text. This work is inspired by similar work in the field of image retrieval, such as Barnard et al. (2003); Carneiro and Vasconcelos (2005). Eck et al. (2008) used boosted classifiers to identify the top k tags describing a particular track, training the classifiers on tags that the users of Last.fm had entered for the track's artist.

There have also been a number of games designed to collect metadata about multimedia. Ellis et al. (2002) described the "Musicseer" system for collecting ground truth about artist similarity, one aspect of which was a game. In this work, players chose which of a list of artists was most similar to a goal artist. The purpose of the game was to link a starting artist to a goal artist with as short a chain of intermediate similar artists as possible. By performing this forced choice of the most similar artist from a list, triplets of relative



(a) ListenGame (b) Tag a Tune (c) MoodSwings

Figure 4: Screenshots from current human computation games for collecting music metadata

similarity were collected, which could then be used to infer a full similarity matrix. (von Ahn and Dabbish, 2004) described the "ESP Game"[4], which asks pairs of players to describe the same image. Once both players agree on a word, they score a certain number of points and move on to the next image. The players attempt to agree on as many images as possible within a time limit. While previous data-collection games maintained data integrity by forcing players to choose from predefined responses, this was the first game to popularize the idea of allowing any response, provided it was verified by a second player.

1.3 Recent music games

Inspired by the success of the ESP game, a number of music-related *human computation* games have been developed. These games aim to provide both entertainment for the players and useful metadata about music for the game operator. Figure 4 shows screenshots from these games and Table 1 compares various aspects of these games and MajorMiner.

The ListenGame[5] (Turnbull et al., 2007b) is a communal game in which a group of players simultaneously selects the best and worst words from six choices to describe a 15-second clip from a song in the CAL-250 dataset (Turnbull et al., 2007a), a set of 250 popular songs from many genres. Players are rewarded in proportion to the number of other users agreeing with their choices. The six choices all come from the same category in a pre-defined taxonomy of tags, e.g. instrumentation, usage, genre. The game also includes a bonus round where players can suggest a new word to be added to a particular category.

In Tag a Tune[6] (Law et al., 2007), paired players have to determine whether they are listening to the same song or to different songs. The players enter unconstrained words describing a 30-second clip until they can decide if they are describing the same song. Players are rewarded with a certain amount of points for deciding correctly and with more points for making consecutive correct choices. The songs come from the Magnatune collection of Creative Commons licensed music and can be downloaded after the game. The game also includes a bonus round where players score points by agreeing on which of three clips is least similar to the others.

[4]available at http://www.gwap.com/
[5]http://www.listengame.org/
[6]available at http://www.gwap.com/

	MajorMiner	ListenGame	Tag a Tune	MoodSwings
Music database	(See Table 4)	CAL 250	Magnatune	uspop2002
Synchrony	Asynchronous	Synchronous	Synchronous	Synchronous
Pair/group	Group	Group	Pair	Pair
Task	Agree on tags	Pick best/worst tag	Decide if song is same	Move in synchrony
Data collected	Free text	Multiple choice	Free text & decision	VA trajectory
Users	490	440	—	100
Clips labeled	2300	250	—	1000
Data collected	12,000 verif. tags	26,000 choices	—	1,700 VA traj.
Unique tags	6400	120	—	—

Table 1: Comparison of current human computation games for collecting music metadata. The top half of the table describes aspects of the games design and the bottom half describes the data collected so far. VA stands for valence-arousal space, see Section 1.3 for details. Usage data for Tag a Tune was not available at the time of publication.

The players in MoodSwings[7] (Kim et al., 2008) are also paired, and each traces a trajectory through a continuous two-dimensional space of musical emotion as they listen to the same 30-second clip. The players score points for agreeing with one another, and are awarded more points when their partner moves to agree with them, but their partner's position is only revealed intermittently to avoid biasing the trajectory. As playback of the clip progresses, the two players need to agree more closely to continue scoring points. Musical emotions are described in terms of Thayer's valence-arousal space (Thayer, 1990), in which the valence axis describes the music in terms of positive versus negative emotion and the arousal axis describes the music in terms of high- versus low-energy.

2 MajorMiner game design

We designed the MajorMiner game with many goals in mind, but our main goal was to encourage players to describe the music thoroughly. This goal shaped the design of the rules for scoring. Another important goal, which informed the method for introducing new clips into the game, was for the game to be fun for both new and veteran players. Specifically, new players should be able to score points immediately, and veteran players should be rewarded with opportunities to score additional points.

Other, lesser goals inform the architecture and implementation of the game. The first of these was the avoidance of the "cold start" problem. Specifically, players should be able to score points as soon as possible after the launch of the game, and a single player should be able to play any time he or she wants, without the need for others to be playing at the same time. Another was to avoid the possibility of cheating, collusion, or other manipulations of the scoring system or, worse, the data collected. Our final goal was to make the game accessible to as many people as possible, implementing it as a standard web page, without requiring any special plugins, installation, or setup.

While many games team one player with a single partner, ours, in a sense, teams one player with all of the other players who have ever seen a particular clip. When a player is paired with a single cooperator, it is possible that the two players could be at very different skill levels, have different levels of familiarity

[7] http://schubert.ece.drexel.edu/moodswings/

with the clip under consideration, or even speak different languages, detracting from each player's enjoyment of the game. It is also possible that during times of low usage only one player might be online at a time (although this problem has been addressed in other games by replaying recorded sessions). The non-paired format, on the other hand, allows the most compatible, creative, or expert players to cooperate with each other asynchronously, since an obscure description used by one player will remain available until a second player verifies it. It also provides a more transparent means of introducing new clips into the labeled dataset, as opposed to pretending that a user can score on a new clip when playing against a mostly prerecorded game. These benefits of non-paired games come at the price of vulnerability to asynchronous versions of the attacks that afflict paired games. For example, collusion in a paired game is only possible between players who are paired with each other, but it is possible in a non-paired game between any two players who have seen a particular clip.

2.1 Scoring

The design of the game's scoring rules reflects our main goal: to encourage players to thoroughly describe clips in an original, yet relevant way. In order to foster relevance, players only score points when other players agree with them. In order to encourage originality, players are given more points for being the first to use a particular description on a given clip. Originality is also encouraged by giving zero points for a tag that two other players have already agreed upon.

The first player to use a particular tag on a clip scores two points when it is verified by a second player, who scores one point. Subsequent players do not score any points for repeating that tag. These point allocations (2, 1, 0) need not be fixed and could be changed depending on participation and the rate at which new music is introduced into the game. The number of players who score points by verifying a tag could be increased to inflate overall scoring and point amounts could also be changed to influence the general style of play. We have found, however, that this simple scoring scheme sufficiently satisfies our goal of encouraging players to be thorough. One concern with this system is that later players could be discouraged if all of the relevant descriptions have already been used by two other players. By carefully choosing when clips are shown to players, however, we can avoid this problem and use the tension created by the scoring rules to inspire originality without inducing frustration.

The game has high-score tables, listing the top 20 scoring players over the past day, the past week, and over all time. The principal payoff of the game may be the satisfaction of reaching some standing in these tables. Including the short-term-based tables gives even new players some chance to see their names in lights.

2.2 Picking clips

When a player requests a new clip to describe, we have the freedom to choose the clip any way we want. This freedom allows us to meet our second goal of making the game fun for both new and experienced players. In order for a player to immediately score on a clip, another player must already have seen it. We therefore maintain a pool of clips that have been seen at least once and so are ready to be scored on. For new players, we draw clips from this pool to facilitate immediate scoring. For experienced players, we generally draw clips from this pool, but sometimes pick clips that have never been seen in order to introduce them into the pool. While such clips do not allow a player to score immediately, they do offer the opportunity to be the first to

use many tags, thus scoring more points when others agree.

While clips must be seen by at least one other person to allow immediate scoring, clips that have been seen by many people are difficult to score on. Since more tags are generally verified for clips that have been seen by more people, it becomes increasingly difficult for players to be original in their descriptions. We alleviate this problem by introducing new clips regularly and preferentially choosing clips that have been seen by fewer players from the pool of scorable clips.

In order for players to transition from new to experienced, we define a continuous parameter γ that indicates a players' experience level. On each request for a new clip, an unseen clip is chosen with probability γ and a scorable clip is chosen with probability $1 - \gamma$. When the website was initially launched, γ was defined as the ratio of the number of clips a player had seen to the number of clips anyone had seen. This prevented problems from developing when the pool of scorable clips was small. After many clips were added in this way, this definition of experience became too strict. We have now transitioned to defining an experienced player as one who has listened to more than 100 clips, at which point γ reaches its maximum value of $\frac{1}{3}$. A brand new player has a γ of 0, and γ linearly increases up to that maximum as the player labels more clips.

This scheme for picking clips has a direct impact on the number of times each clip is seen, and hence the overall difficulty of the game. The result, derived in Appendix A:, is that at equilibrium, all clips in the scorable pool will have been seen the same number of times, where that number is the reciprocal of the expected value of γ over all of the players. Computing the expected value of γ from the usage data in Figure 5(b), each clip would have been seen 59 times under the original picking strategy, but will only be seen 7 times under the new strategy.

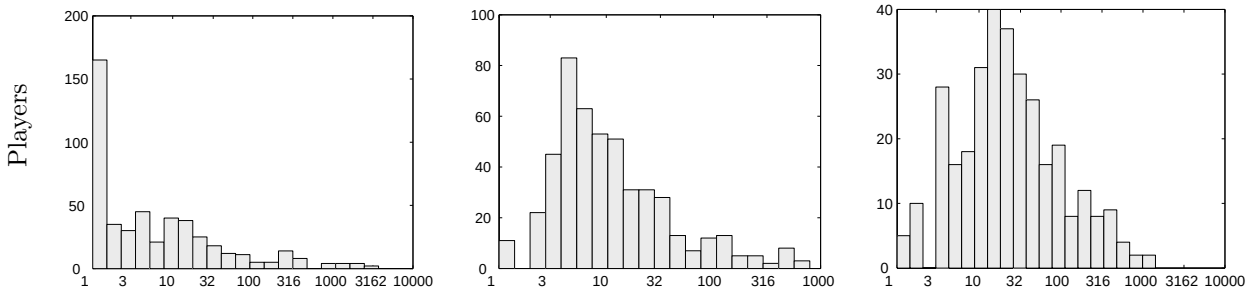
2.3 Revealing labels

Seeing other players' descriptions is part of the fun of the game. It also acclimatizes new players to the words that they have a better chance of scoring with. The other responses can only be revealed after a player has finished labeling a given clip, otherwise the integrity of the data and the scoring would be compromised. With this in mind, we designed a way to reveal other players' tags without giving away too much information or creating security vulnerabilities.

We reveal the tags of the first player who has seen a clip, a decision that has many desirable consequences. This person is uniquely identified and remains the same regardless of how many subsequent players may see the clip. The same tags are thus shown to every player who requests them for a clip and repeated requests by the same player will be answered identically, even if others have tagged the clip between requests. The only person who sees a different set of labels is that first tagger, who instead sees the tags of the second tagger (if there is one).

As described in the previous section, the first player to tag a particular clip is likely to have more experience with the game. Their descriptions are good examples for other players as an experienced player will generally be good at describing clips, will have a better idea of what others are likely to agree with, and will know what sort of formatting to use. Their tags can thus serve as good examples to others.

Also, in order to avoid introducing extra-musical context that might bias the player, we only reveal the name of the artist, album, and track after a clip is finished being labeled. This focuses the player much more on describing the sounds immediately present and less on a preconception of what an artist's music sounds like. It is also interesting to listen to a clip without knowing the artist and then compare the sound to the



(a) Score (b) Clips heard (c) Vocabulary

Figure 5: Histograms of player data. Y-axis is number of players, x-axis is the specified statistic in logarithmic units. (a) Number of points each player has scored, (b) Number of clips each player has listened to, (c) Number of unique tags each player has used.

preconceptions one might have about the artist afterward.

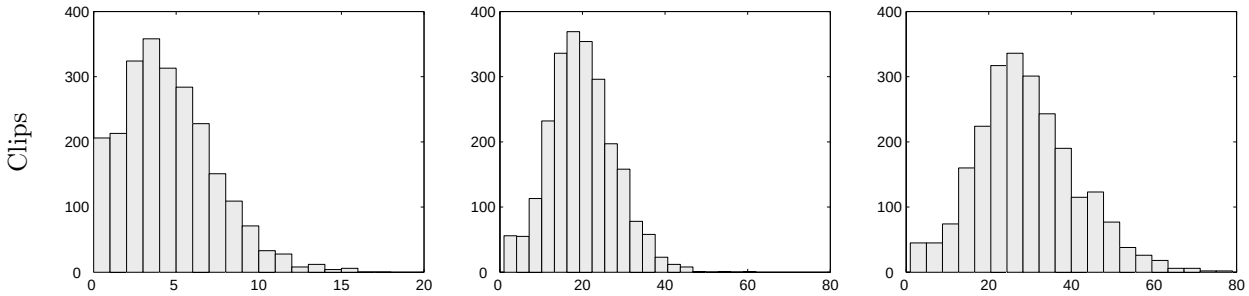
2.4 Strategy

When presented with a new clip, a player does not know which tags have already been applied to it. Trying one of the more popular tags will reveal how many times that tag has been used and thus the approximate number of times the clip has been seen. If the popular tag has never been used or has been used only once, the player can apply other popular tags and collect points relatively easily. If the tag has already been used twice, however, it is likely to be more difficult to score on the clip. The player must then decide whether to be more original or go on to another clip.

This clip-wise strategy leads to two overall strategies. The first is to be as thorough as possible, scoring points both for agreeing with existing tags and by using original tags that will later be verified. By agreeing with existing tags, the thorough player both collects single-point scores and prevents future listeners from scoring on those tags. By using original tags, the thorough player will setup many two-point scores when subsequent players encounter the same clip. The second strategy is to listen to as many clips as possible, trying to use popular tags on clips that haven't been seen before.

While having a large number of clips with popular labels is worthwhile, in-depth analysis is more useful for us. To encourage breadth of description, we could add a cost to listening to clips or to posting tags, which would motivate players to use tags they were more certain would be verified by others. Similarly, we could post the high scores in terms of the number of points scored per clip heard or the number of points scored per tag. These adjustments to the scoring philosophy would encourage players to be more parsimonious with their tagging and listening. We have not yet encountered the shallow, speedy strategy and so have not instituted such measures.

We have guarded against a few possible exploits of the system. The first is collusion between two players, or the same person with two different usernames. Two players could, in theory, communicate their tags for particular clips to each other and score on all of them. We thwart this attack by making it difficult for players to see clips of their choosing and by adding a refractory period between presentations of any particular clip. Since players can only see their most recent clips, we also never refer to clips by an absolute identifier, only by relative positions in the recently seen and recently scored lists, making it more difficult to memorize which



(a) Scored tags (b) Unique tags (c) All tags

Figure 6: Histograms of clip data. Y-axis is number of clips, x-axis is the specified statistic. (a) Number of tags that have been used by at least two players on each clip, (b) Number of unique tags that have been applied to a clip, and (c) Number of tags that have been applied to a clip.

clips have been seen. Another potential exploit of the system is an extreme form of the speedy strategy in which a player repeatedly uses the same tag or tags on every clip, regardless of the music. This is easily detected and can be neutralized by disabling the offending account.

3 Data collected

At the time of this paper's writing, the site has been live for 11 months, in which time 489 players have registered. A total of 2308 clips have been labeled, being seen by an average of 6.99 players each, and described with an average of 31.24 tags each, 5.08 of which have been verified. See Table 2 for some of the most frequently used descriptions and Figures 5 and 6 for histograms of some statistics of player and clip data, respectively.

The system was implemented as a web application using the Ruby on Rails framework. The player needs only a browser and the ability to play mp3s, although javascript and flash are helpful and improve the game playing experience. The page and the database are both served from the same Pentium III 733 MHz with 256MB of RAM. This rather slow server can sustain tens of simultaneous players.

The type of music present in the database affects the labels that are collected, and our music is relatively varied. By genre, it contains electronic music, indie rock, hip hop, pop, country, mainstream contemporary rock, and jazz. Much of the music is from independent or more obscure bands, which diminishes the biases and extra context that come from the recognition of an artist or song. See Table 4 for the tags that users of Last.fm have applied to this music collection.

Those 2308 clips were selected at random from a collection of 97060 clips, which exhaustively cover 3880 tracks without overlap. The clips that were selected came from 1441 different tracks on 821 different albums from 489 different artists. This means that the average artist had 4.7 clips tagged, the average album had 2.8 clips tagged, and the average track had 1.6 clips tagged. The most frequently seen items, however, had many more clips tagged. Saint Etienne had 35 of their clips tagged, Kula Shaker's album *K* had 12 of its clips tagged, and Oasis' track "Better Man" had 9 of its clips tagged.

Certain patterns are observable in the collected descriptions. As can be seen in Table 2, the most popular tags describe genre, instrumentation, and the gender of the singer, if there are vocals. People use descriptive

Label	Verified	Used	Players
drums	908	3032	114
guitar	837	3146	176
male	723	2450	95
rock	658	2614	198
electronic	484	1855	127
pop	477	1755	148
synth	471	1770	85
bass	417	1632	99
female	342	1387	100
dance	321	1242	115
techno	244	933	101
piano	179	826	120
electronica	167	679	66
vocal	163	824	76
synthesizer	162	681	49
slow	157	726	90
rap	151	723	129
voice	140	735	50
hip hop	139	535	97
jazz	129	696	149
vocals	128	704	50
beat	125	628	76
80s	111	488	69
fast	109	494	70
instrumental	102	536	62

Table 2: The 25 most popular tags in the MajorMiner game. Three measures of tag popularity are provided: the number of clips on which the tag was verified by two players, the total number of times the tag was used, including unverified uses and uses by more than two players, and the number of players who have ever used the tag.

words, like `soft`, `loud`, `quiet`, `fast`, `slow`, and `repetitive`, but do so less frequently. Emotional words are even less popular, perhaps because they are difficult to verbalize in a way that others will likely agree with. There are hardly any words describing rhythm, except for an occasional `beat` tag.

Since any tag is allowed, players can and do use the names of artists they recognize. For example, `u2` has been verified 15 times, `depeche mode` 12 times, and `bowie` 8 times. Only five of the clips verified as `bowie` were actually performed by David Bowie, however, the other three were performed by Gavin Friday, Suede, and Pulp. One need not take these descriptions literally; they could, for instance, be indicating a similarity between the particular clip from Suede's song "New Generation" and some aspect or era of David Bowie's music. These comparisons could indicate artists who are good candidates for the anchors in an anchor space of artist descriptions (Berenzweig et al., 2003). Such a system would describe new artists by their musical relationships to well known artists.

Another valid and easily verified description of a clip is its lyrical content, if decipherable. Ten seconds are generally enough to include a line or two of lyrics, which the player then must distill down to one or two words. This has the added benefit of isolating some of the more important words from the lyrics, since players want to make their descriptions easy for others to match. Currently, `love` seems to be the most popular lyric word, with 19 verified uses.

The top tags in the game have been quite stable since it began. We would like to believe that this stability results from their being the most appropriate words to describe the music in our database. It is possible, however, that the first players' choices of words had a disproportionate impact on the vocabulary of the game.

This might have happened through the feedback inherent in the point scoring system and in the revealing of players' descriptions. If the first players had used different words, those words might now be the most popular. It would not be difficult to divide the players between separate game worlds in order to test this hypothesis, although we do not currently have enough players to attempt this.

3.1 Data normalization

In the initial implementation of the system, tags only matched when they were identical to each other. This was too strict a requirement, as `hip hop` should match `hip-hop` in addition to misspellings and other variations in punctuation. Since we still had all of the tagging data, however, it was possible to perform an offline analysis of the tags, i.e. replay the entire history of the game, to compare the use of different matching metrics. Below, we describe the metric that we settled on for the matching in the previously collected data. After experimenting on the existing data, we implemented a similar scheme in the live game website and re-scored all of the previous game-play.

Our offline data analysis consisted of a number of manual and semi-supervised steps. We began with 7698 unique tags. First, we performed a spell check on the collection of tags, in which misspellings were corrected by hand, reducing the total number of tags to 7360. Then, we normalized all instances of `&`, `and`, `'n'`, etc, leaving us with 7288 tags. Next, we stripped out all of the punctuation and spaces, reducing the collection to 6987 tags. And finally, we stemmed the concatenated tag, turning plural forms into singular forms and removing other suffixes, for a final count of 6363 tags, a total reduction of 1335 duplicate tags. The first steps generally merged many unpopular tags with one popular tag, for example all of the misspellings of `synthesizer`. The later steps tended to merge two popular tags together, for example `synth` and `synthesizer`. Merging two popular tags generally affects the scoring of the game overall, while merging orphans improves the game playing experience by making the game more forgiving.

Because of the way the rules of the game are defined, merging tags affects the score for a particular clip, increasing it when a single player used each version, but possibly decreasing it if both versions had already been verified. Not only does merging tags affect the score on a particular clip, but it also affects the total number of matching tags in the game. A net increase in scoring means that merging two tags was beneficial in uniting players with slightly different vocabularies, while a net decrease in scoring means that players might have been using two tags to try to increase their scores without conveying new information. See Table 3 for examples of the tags that were affected the most by merging and how much their merging affected the overall number of verified tags.

4 Autotagging experiments

Autotagging is the process of automatically applying relevant tags to a musical excerpt. While the system of Eck et al. (2008) selects the best k tags to describe a clip, we pose the problem as the independent classification of the appropriateness of each potential tag. Each tag is generally only applied to a small fraction of the clips, so in this version of autotagging there are many more negative than positive examples. Since the support vector machines (SVMs) we use are more effective when both classes are equally represented in the training data, we randomly select only enough negative examples to balance to positive examples. For

Tags	Initial	Merged	Net change
vocal, vocals	163+128	355	+64
hip hop, hiphop, hip-hop	139+29+22	166	+24
drum and bass, drum n bass, ...	16+6+...	48	$\sim$ +24
beat, beats	125+9	154	+20
horn, horns	27+15	48	+15
drum, drums	908+54	962	0
80s, 80's	111+28	130	−9
synth, synthesizer, synthesizers	471+162+70	498	−205

Table 3: Number of verified uses of each tag before and after merging slightly different tags with each other. Merging tags can lead to a reduction in verified uses when several separately-verified variants of a term are merged, leaving just a single verified use.

testing, we similarly balance the classes to provide a fixed baseline classification accuracy of 0.5 for random guessing.

To compare different sets of tags, we use the system described by Mandel and Ellis (2008), which was submitted to the MIREX evaluations (Downie et al., 2005) on music classification in 2007. While the results of such experiments will certainly vary with the details of the features and classifier used and many music classification systems have been described in the literature (e.g. Lidy et al., 2007; Pachet and Roy, 2007; Turnbull et al., 2006; Eck et al., 2008), these experiments are meant to provide a lower bound on the amount of useful information any autotagging system could extract from the data. Furthermore, our system is state-of-the-art, having achieved the highest accuracy in the MIREX 2005 audio artist identification task and performed among the best in the MIREX 2007 audio classification tasks.

The system primarily uses the spectral features from Mandel and Ellis (2005), but also uses temporal features that describe the rhythmic content of each of the clips. For each tag, these features were classified with a binary support vector machine using a radial basis function kernel. The performance of the classifiers was measured using 3-fold cross-validation, in which $\frac{2}{3}$ of the data was used for training and $\frac{1}{3}$ for testing. Since some of the tags had relatively few positive examples, we repeated the cross-validation experiment for 3 different random divisions of the data, increasing the total number of evaluations to 9. We ensured the uniform size of all training and test sets for classifiers that were being compared to each other. When there were more positive or negative clips than we needed for a particular tag, the desired number was selected uniformly at random.

We used two sets of tags from MajorMiner. The first consists of all of the (clip, tag) pairs that had been verified by at least two people as being appropriate. We call this dataset "MajorMiner verified." The second consists of all of the (clip, tag) pairs that were submitted to MajorMiner. We call this dataset "MajorMiner all." The game is designed to collect the verified dataset, and the inclusion of the complete dataset and its poorer performance shows the usefulness of tracking when tags are verified.

4.1 Social tag data

Using the audioscrobbler web API, we collected all of the social tag data that users of the website Last.fm had applied to MajorMiner's corpus of music. We call these "social" tags because there is a great deal of communal collaboration that goes into applying them, much of which occurs on the social networks of music

Last.fm 25	Clips	Last.fm 50	Clips	Last.fm 75	Clips
rock	1229	albums I own	816	albums I own	688
alternative	1178	electronic	765	indie	552
indie	992	rock	735	rock	526
albums I own	935	indie	729	electronic	481
electronic	871	alternative	507	90s	341
indie rock	668	90s	473	alternative	287
90s	608	pop	345	pop	272
pop	591	britpop	255	britpop	223
electronica	542	female vocalist	244	mistagged artist	187
80s	517	80s	226	Hip-Hop	185
british	513	idm	209	new wave	140
female vocalists	479	new wave	208	female vocalist	138
britpop	377	electronica	199	singer-songwriter	117
alternative rock	354	Hip-Hop	192	classic rock	104
trip-hop	318	indie rock	187	jazz	103
jazz	303	mistagged artist	187	post-punk	103
classic rock	302	oldies	157	00s	98
idm	276	singer-songwriter	151	80s	93
downtempo	263	00s	146	shoegaze	93
trance	238	british	138	trip-hop	92
new wave	229	trip-hop	133	electronica	83
dance	227	post-punk	131	indie rock	76
soul	216	ambient	125	collected	75
00s	214	jazz	122	ambient	70
Hip-Hop	211	collected	121	british	68

Table 4: Top 25 tags describing our music from Last.fm with a "count" of at least 25, 50, or 75. Each tag is given with the number of clips it has been associated with out of the 2308 total clips.

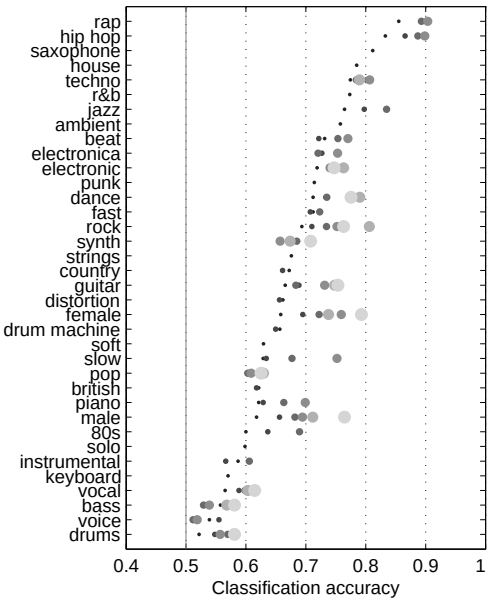


Figure 7: Mean classification accuracy for the top 30 verified tags from MajorMiner using different amounts of training data. Larger markers indicate larger amounts of training data: 33, 48, 68, 98, 140, 200 positive examples with an equal number of negative examples. The test sets all contained 10 positive and 10 negative examples, which means that for these plots $N = 180$, and a difference of approximately 0.07 is statistically significant under a binomial model.

websites like Last.fm (Eck et al., 2008). Last.fm users can tag artists, albums, and tracks, but not clips. To label a clip with these data, then, we used the tags that had been applied to its source track along with that track's album and artist. Using artist and album tags is necessary because many tracks have not been tagged much or even at all on Last.fm. The music in our corpus was generally well-tagged on Last.fm: only 15 out of 489 artists had fewer than 5 tags associated with them, as did 357 of 821 albums, and 535 of 1441 tracks. By combining tags from these three sources, we have at least one tag for 2284 of our 2308 clips. Of these, 1733 clips have at least one track tag, 1684 have at least one album tag, and 2284 have at least one artist tag.

Last.fm supplies tags with a "count" parameter between 0 and 100, representing how much that tag applies to a particular item. While the exact algorithm used to calculate this count is not publicly available, it appears to be similar to a term-frequency, inverse document frequency measure (TF-IDF). We created three different datasets by thresholding these counts at three different values, 25, 50, and 75, which we call Last.fm 25, Last.fm 50, and Last.fm 75, respectively. We counted a tag as having been applied to a clip when it had at least the threshold count for that clip's artist, album, or track. See Table 4 for a list of the most popular tags from each of these three datasets and the number of clips associated with each. In comparing this to Table 2, it can be seen that the Last.fm and MajorMiner tag sets share many concepts, although they vary in many particulars.

After normalizing these tags in much the same way as the MajorMiner tags, we were left with 477 unique tags at the 75-count threshold, 751 at the 50-count threshold, and 1224 at the 25-count threshold. The tags from Last.fm are in general genre-related, although there are some non-genre tags like `90s`, `albums I own`, and `female vocalist`. The tags from MajorMiner contain many genre terms, but also terms about the music's sonic characteristics and its instrumentation.

4.2 Classification with MajorMiner data

Our first experiment measured the effect of varying the amount of training data on classification accuracy. We first evaluated all of the MajorMiner tags that had been verified at least 50 times, sampling 50 clips for those tags that had been verified on more than 50 clips. Since we used a three-way cross-validation, this means that approximately 33 positive examples and 33 negative examples were used for training on each fold. The results can be seen as the smallest markers in Figure 7 and the tags are sorted by their average classification accuracy on this task. Notice that even using 33 positive examples, results are quite good for tags such as `rap`, `house`, `jazz`, and `electronica`.

In general, classification accuracy was highest for genres and lowest for individual instruments. This makes sense because the spectral features we use in the classifiers describe overall timbre of the sound as opposed to distinct instruments. One exception to this is `saxophone`, which can be identified quite accurately. This anomaly is explained by the tag's strong correlation with the genre `jazz`. Tags with intermediate performance include descriptive terms such as `dance`, `distortion`, and `fast`, which are classified relatively well.

As the amount of training data increased, so too did classification accuracy. For tags like `male`, `female`, `rock`, `dance`, and `guitar` an increase in training data from 33 to 200 tags improved accuracy by 10–20 percentage points. This trend is also evident, although less pronounced, in less popular tags like `slow`, `80s`, and `jazz`. No tag performed significantly worse when more training data was available, although some performed no better.

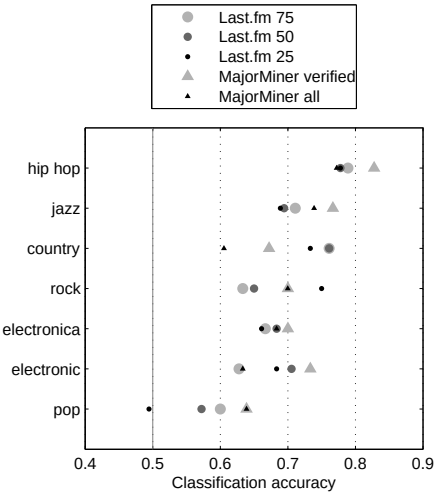


Figure 8: Comparison of classification accuracy between data sets for tags that appeared in all of them. In the evaluation, $N = 180$, so for the accuracies encountered a difference of approximately 0.07 is statistically significant under a binomial model.

4.3 Direct comparison with social tags

Certain tags were popular in all of the MajorMiner and Last.fm datasets and we can directly compare the accuracy of classifiers trained on the examples from each one. See Figure 8 for the results of such a comparison, in which the tags are sorted by average classification accuracy. Each of these classifiers was trained on 28 positive and 28 negative examples from a particular dataset. For five out of these seven tags the verified MajorMiner tags performed best, but on the `rock` and `country` tags the classifiers trained on Last.fm 25 and Last.fm 75, respectively, were approximately 8 percentage points more accurate. The verified MajorMiner tags were always classified more accurately than the complete set of MajorMiner tags. Of the three Last.fm datasets, Last.fm 75 just barely edges out the other two, although their accuracies are quite similar.

This performance difference can be attributed to three sources of variability. The first is the amount that the concept being classified is captured in the features that are input to the classifier. This variation is mainly exhibited as the large differences in performance between tags, as some concepts are more closely tied to the sound than others and of those some are more closely tied to sonic characteristics that are captured in our features. While this source of variability is present in both MajorMiner and Last.fm data, we believe that MajorMiner's tags are more sonically relevant because they are applied to clips instead of larger musical elements. Also, the Last.fm datasets contain some extra-musical tags like `albums I own`, which are generally not discernible from the sound.

The second source of variability is inter-subject variability, caused by differences between individuals' conceptions of a tag's meaning. Because of its collaborative nature, MajorMiner promotes agreement on tags. Last.fm also promotes such agreement through its "count" mechanism, which scores the appropriateness of a tag to a particular entity. The difference in performance between classifiers trained on the two MajorMiner datasets shows that verified instances of tags are more easily predicted. A comparison of the Last.fm tags from the three different thresholds shows some evidence that tags with a higher "count" are more easily learned by classifiers, although the overall differences are minor.

Dataset	Mean	Std
MajorMiner verified	0.672	0.125
MajorMiner all	0.643	0.109
Last.fm 50	0.626	0.101
Last.fm 25	0.624	0.104
Last.fm 75	0.620	0.111

Table 5: Overall mean and standard deviation classification accuracy for each dataset on its 25 most prevalent tags. All classifiers were trained on 40 positive and 40 negative examples of each tag.

The final source of variability is intra-subject variability, caused by the inconsistencies in an individual's conception of a tag's meaning across multiple clips. This source of variability is present in all datasets, although it can be mitigated by using expert labelers, who have better-defined concepts of particular musical descriptors and are thus more consistent in their taggings. Even though no expert knows all areas of music equally well, it should be possible to construct a patchwork of different experts' judgments that is maximally knowledgeable. MajorMiner's non-paired (or one-paired-with-all) game-play allows experts in different types of music to collaborate asynchronously. As long as they use a distinct vocabulary, it will also select their areas of expertise automatically. It is not clear from our data how many such experts have played the game, but we suspect that this phenomenon will emerge with greater participation.

4.4 Overall performance

Finally, we compared the accuracy of classifiers trained on the top 25 tags from each of the datasets. The overall mean accuracy along with its standard deviation can be seen in Table 5. The variance is quite high in those results because it includes inter-tag and intra-tag variation, inter-tag being the larger of the two. For a breakdown of performance by tag and cross-validation fold, see Figure 9, from which it is clear that the variance is generally quite low, although exceptions do exist.

While these tags are by no means uncorrelated with one another, we believe that it is meaningful to average their performance as they are separate tokens that people chose to use to describe musical entities. While many of MajorMiner's players might consider `hip hop` and `rap` to be the same thing, they are not perfectly correlated, and certain clips are more heavily tagged with one than the other. For example, while the track shown in Figure 1 might be considered to fall entirely within the hip hop genre, only certain sections of it include rapping. Those sections are particularly heavily tagged `rap`, while the rest of the song is not.

Overall, the average classification accuracy was highest for the top 25 verified tags from MajorMiner. The fact that verified tags make better training data than all tags supports the reasoning behind our design of the scoring rules. Thresholding the "count" of a tag made little difference in the mean accuracy of the Last.fm datasets, although there were differences between the datasets' performances on specific tags. For low thresholds, there are a small number of tags that perform quite well, while most perform poorly. As the threshold is increased, there are fewer stand-out tags and the classification becomes more consistent across tags.

The top 25 tags for our clips on Last.fm do not include musical characteristics or instruments, but do include some extra-musical tags like `albums I own`, `indie`, `british`, and `90s`. Such tags are not well classified by our system because they have less to do with the sound of the music than with the cultural

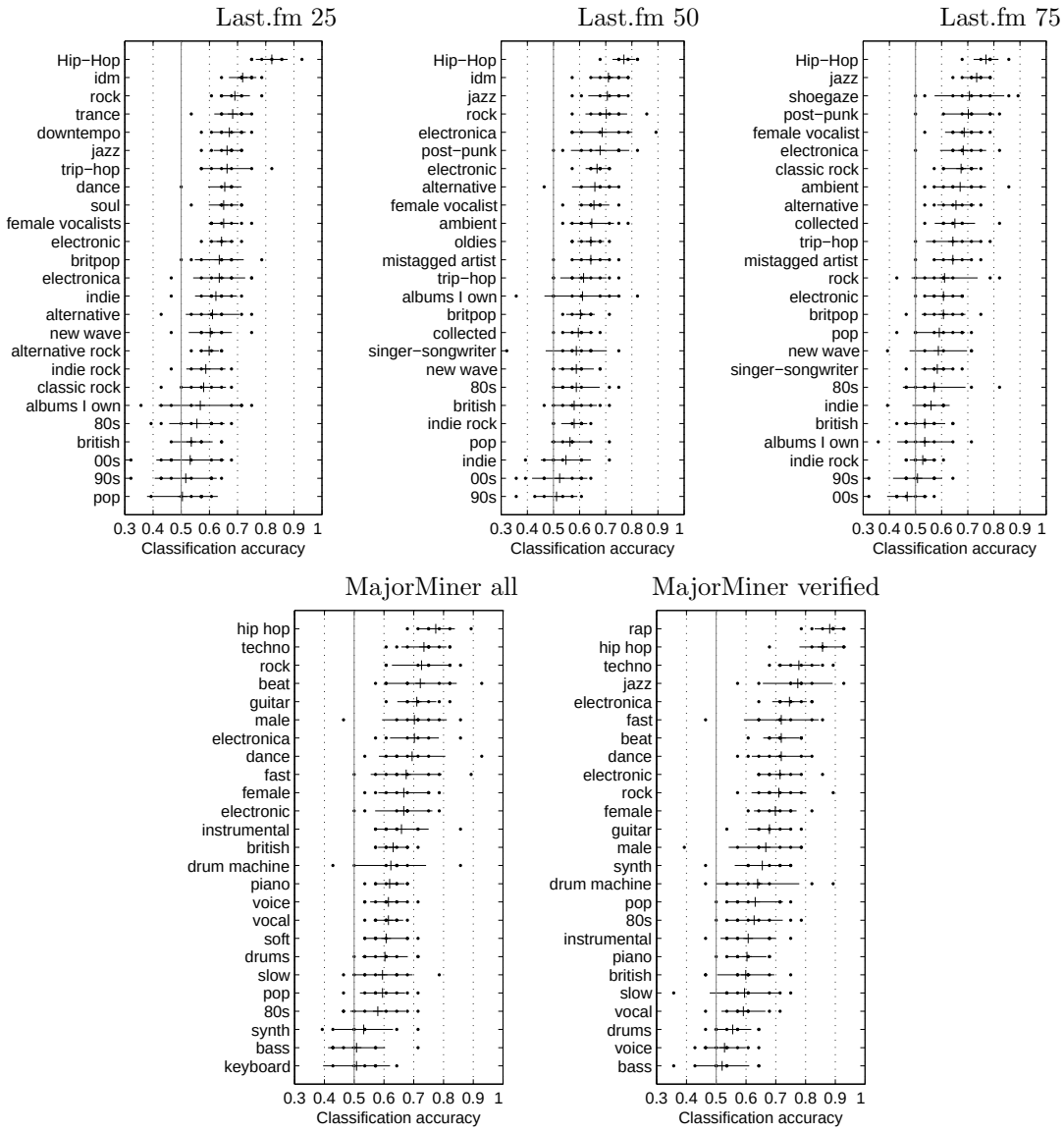


Figure 9: Classification accuracy for the top 25 tags from each dataset. Dots are individual cross-validation folds, plus signs are means, and the horizontal bars are standard deviations. The vertical gray line indicates chance performance, 0.5. In the evaluation, $N = 252$, so under a binomial model an accuracy of > 0.563 is significantly better than random (0.5).

context surrounding it. Exceptions do arise because of correlations in the data, for example, Last.fm's `albums I own` is highly correlated with `rock` in the MajorMiner data. Such correlations could help associate tags with no apparent sonic relevance to tags that have a firmer acoustic foundation.

5 Conclusions

We have presented a game for collecting objective, specific descriptions of musical excerpts. Playing the game is, in our estimation, fun, interesting, and thought provoking. Preliminary data collection has shown that it is useful for gathering relevant, specific data and that players agree on many characteristics of clips of music. Experiments show that these tags are useful for training automatic music description algorithms, more so than social tags from a popular website.

We believe that audio classifiers are more successful at learning the MajorMiner tags because they are more closely tied to the sound of the music itself. Social tags, on the other hand, also include many non-musical factors, especially when applied to artists and albums. Even though they are more noisy, however, social music websites like Last.fm have proven very popular and have amassed billions of taggings, more than a game like MajorMiner could ever hope to collect. This tradeoff between tag quality and quantity implies that the two sources of data are useful in different circumstances. For example, they could be combined in a hybrid approach that builds noise robustness on the high quality data before exploring the noisy, high quantity data.

There is much that still remains to be done with this system. Among other things, we would like to investigate ways to combine audio-based and word-based music similarity to help improve both, to use automatic descriptions as features for further manipulation, to investigate an anchor space built from the data collected here, and to use descriptions of clips to help determine the structure of songs.

Acknowledgments

The authors would like to thank Johanna Devaney for her help and Douglas Turnbull, Youngmoo Kim, and Edith Law for information about their games. This work was supported by the Fu Foundation School of Engineering and Applied Science via a Presidential Fellowship, by the Columbia Academic Quality Fund, and by the National Science Foundation (NSF) under Grants IIS-0238301 and IIS-0713334. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Appendix A: Derivation of clip seen distribution

Section 2.2 sets out a simple rule for choosing clips to present the a player: with probability γ a clip is selected that no other player has seen before. Otherwise, the scorable clip that has been seen by the fewest number of other players is selected. The way in which γ grows with user experience induces a distribution over the number of times a clip will be seen. In designing this growth, it is useful to have a model of this relationship and we can create this model from a probabilistic formulation of user experience and past useage data from the game.

To perform this calculation, we define a Markov chain with states $n \in \mathbb{N}$. Clips move around in this state space such that a clip's state represents the number of users that have seen that clip. Let x_n be the number of clips that are in state n, i.e. the number of clips that have been seen n times. The system has $x_0 = \infty$ and begins with all other $x_n = 0$. When player i requests a clip, a coin is flipped and with probability γ_i the player is given a new clip, which moves from state 0 to state 1. Otherwise the player is given the scorable clip that has been seen by the fewest other players, moving it from n^* to $n^* + 1$ where $n^* > 0$ is the lowest populated state.

Assume for the moment that all users have the same probability of receiving a new clip, i.e. $\gamma_i = \gamma, \forall i$. Then at equlibrium, only two states are occupied, $\bar{n} \equiv \left\lfloor \frac{1}{\gamma} \right\rfloor$ and $\bar{n} + 1$. The occupancy of state $\bar{n}$ is $x_{\bar{n}} = N\left(\bar{n} - \frac{1-\gamma}{\gamma}\right)$ and the occupancy of state $\bar{n}+1$ is $x_{\bar{n}+1} = N - x_{\bar{n}}$. This holds true even when $\gamma = E[\gamma_i]$, where the expectation is taken over the true distribution of γ_i values seen in the game. In this case, at equilibrium, $\bar{n} = \left\lfloor \frac{1}{E[\gamma_i]} \right\rfloor$ and all clips will have been seen either $\bar{n}$ or $\bar{n} + 1$ times. In quantifying a user's experience level, we are defining the function $\gamma_i = \gamma(c_i)$, where c_i is the number of clips player i has already heard.

This analysis assumes that every player is equally likely to select the next clip, i.e. that there is no correlation between the players that see successive clips. In general, this is not the case, as one person will generally play for a certain amount of time, creating a burst of clips seen by a single player. The analysis does, however, describe the gross effects of these picking strategies.

References

K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D.M. Blei, and M.I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3(6):1107–1135, 2003.

A. Berenzweig, D. P. W. Ellis, and S. Lawrence. Anchor space for classification and similarity measurement of music. In *Proc Intl Conf on Multimedia and Expo (ICME)*, 2003.

G. Carneiro and N. Vasconcelos. Formulating Semantic Image Annotation as a Supervised Learning Problem. In *Computer Vision and Pattern Recognition*, volume 2, pages 163–168, 2005.

S. J. Downie, K. West, A. Ehmann, and E. Vincent. The 2005 music information retrieval evaluation exchange (MIREX 2005): Preliminary overview. In J. D. Reiss and G. A. Wiggins, editors, *Proc. Intl. Symp. Music Information Retrieval*, pages 320–323, September 2005.

D. Eck, P. Lamere, T. B. Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, 2008.

D. P. W. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proc. Intl. Symp. Music Information Retrieval*, 2002.

Y. E. Kim, E. Schmidt, and L. Emelle. MoodSwings: A collaborative game for music mood label collection. In *Proc. Intl. Symp. Music Information Retrieval*, September 2008. To appear.

E. L. M. Law, L von Ahn, R. B. Dannenberg, and M. Crawford. TagATune: A game for music and sound annotation. In S. Dixon, D. Bainbridge, and R. Typke, editors, *Proc. Intl. Symp. Music Information Retrieval*, pages 361–364, 2007.

T. Lidy, A. Rauber, A. Pertusa, and J. M. Iñesta. Improving genre classification by combination of audio and symbolic descriptors using a transcription system. In S. Dixon, D. Bainbridge, and R. Typke, editors, *Proc. Intl. Symp. Music Information Retrieval*, September 2007.

M. I. Mandel and D. P. W. Ellis. Song-level features and support vector machines for music classification. In J. D. Reiss and G. A. Wiggins, editors, *Proc. Intl. Symp. Music Information Retrieval*, pages 594–599, September 2005.

M. I. Mandel and D. P. W. Ellis. Multiple-instance learning for music information retrieval. In *Proc. Intl. Symp. Music Information Retrieval*, September 2008. To appear.

F. Pachet and P. Roy. Exploring billions of audio features. In *International Workshop on Content-Based Multimedia Indexing(CBMI)*, pages 227–235, 2007.

R. E. Thayer. *The Biopsychology of Mood and Arousal*. Oxford University Press, USA, September 1990.

D. Turnbull, L. Barrington, and G. Lanckriet. Modeling music and words using a multi-class naive bayes approach. In *Proc. Intl. Symp. Music Information Retrieval*, October 2006.

D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query-by-semantic description using the CAL500 data set. In *ACM Special Interest Group on Information Retrieval (SIGIR)*, 2007a.

D. Turnbull, R. Liu, L. Barringon, and G. Lanckriet. A game-based approach for collecting semantic annotations of music. In S. Dixon, D. Bainbridge, and R. Typke, editors, *Proc. Intl. Symp. Music Information Retrieval*, pages 535–538, September 2007b.

L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proc SIGCHI conference on Human factors in computing systems*, pages 319 – 326, 2004.

B. Whitman and D. Ellis. Automatic record reviews. In *Proc. Intl. Symp. Music Information Retrieval*, 2004.

B. Whitman and R. Rifkin. Musical query-by-description as a multiclass learning problem. In *IEEE Workshop on Multimedia Signal Processing*, pages 153–156, 2002.