FIVE APPROACHES TO COLLECTING TAGS FOR MUSIC

Douglas Turnbull
UC San Diego
dturnbul@cs.ucsd.edu

Luke Barrington
UC San Diego
lbarrington@ucsd.edu

Gert Lanckriet
UC San Diego
gert@ece.ucsd.edu

ABSTRACT

We compare five approaches to collecting tags for music: conducting a survey, harvesting social tags, deploying annotation games, mining web documents, and autotagging audio content. The comparison includes a discussion of both scalability (financial cost, human involvement, and computational resources) and quality (the cold start problem & popularity bias, strong vs. weak labeling, vocabulary structure & size, and annotation accuracy). We then describe one state-of-the-art system for each approach. The performance of each system is evaluated using a tag-based music information retrieval task. Using this task, we are able to quantify the effect of popularity bias on each approach by making use of a subset of more popular (short-head) songs and a set of less popular (long-tail) songs. Lastly, we propose a simple hybrid context-content system that combines our individual approaches and produces superior retrieval results.

1 INTRODUCTION

Tags are text-based tokens, such as "happy", "classic rock", and "distorted electric guitar", that can be used to annotate songs. They represent a rich source of semantic information that is useful for text-based music retrieval (e.g., [19]), as well as recommendation, discovery, and visualization [11]. Tags can be collected from humans using surveys [19, 5], social tagging websites [13], or music annotation games [20, 14, 12]. They can also be generated by text mining web-documents [10, 24] or by autotagging audio content [19, 7, 21]. In Section 2, we introduce key concepts associated with tag collection, and use them to highlight the strengths and weaknesses of each of these five approaches. In Section 3, we describe one implementation of a system for each approach and evaluate its performance on a tag-based music retrieval task. In the final section, we describe a simple hybrid system that combines the output from each of our individual systems.

2 COLLECTING TAGS

In this section, we describe five approaches to collecting music tags. Three approaches (surveys, social tags, games) rely on human participation, and as such, are expensive in terms of financial cost and human labor. Two approaches (text mining, autotagging) rely on automatic methods that are computationally intense, but require less direct human involvement.

There are a number of key concepts to consider when comparing these approaches. The *cold start problem* refers to the fact songs that are not annotated cannot be retrieved. This problem is related to *popularity bias* in that popular songs (in the *short-head*) tend to be annotated more thoroughly than unpopular songs (in the *long-tail*) [11]. This often leads to a situation in which a short-head song is ranked above a long-tail song despite the fact that the long-tail song may be more semantically relevant. We prefer an approach that avoids the cold start problem (e.g., autotagging). If this is not possible, we prefer approaches in which we can explicitly control which songs are annotated (e.g., survey, games), rather than an approach in which only the more popular songs are annotated (e.g., social tags, web documents).

A *strong labeling* [3] is when a song has been explicitly labeled or not labeled with a tag, depending on whether or not the tag is relevant. This is opposed to a *weak labeling* in which the absence of a tag from a song does not necessarily indicate that the tag is not relevant. For example, a song may feature drums but is not explicitly labeled with the tag "drum". Weak labeling is a problem if we want to design a MIR system with high recall, or if our goal is to collect a training data set for a supervised autotagging system that uses discriminative classifiers (e.g., [7, 24]).

It is also important to consider the size, structure, and extensibility of the tag vocabulary. In the context of text-based music retrieval, the ideal vocabulary is a large and diverse set of semantic tags, where each tag describes some meaningful attribute or characterization of music. In this paper, we limit our focus to tags that can be used consistently by a large number of individuals when annotating novel songs based on the audio content alone. This does not include tags that are personal (e.g., "seen live"), judgmental (e.g., "horrible"), or represent external knowledge about the song (e.g., geographic origins of an artist). It should be noted that these tags are also useful for retrieval (and recommendation) and merit additional attention from the MIR community.

A tag vocabulary can be *fixed* or *extensible*, as well as *structured* or *unstructured*. For example, the tag vocabulary associated with a survey can be considered fixed and structured since the set of tags and the grouping of tags into coherent semantic categories (e.g., genres, instruments, emotions, usages) is predetermined by experts using domain knowledge [19, 20]. By contrast, social tagging communities produce a vocabulary that is extensible since any user can

Approach	Scalability / Cost			Quality			
	Financial	Human	Computational	'Cold Start'	Labeling	Vocabulary	Accuracy
Survey	Expensive	Expensive	**Cheap**	Decent	**Strong**	Structured	**Great**
Pandora	$20-$30 / survey	20 min / survey	Tools, DB	Long Backlog		400 Tags	Professional
CAL500	$\sim$ $1 / survey	6 min / survey	Tools, DB	Small Sample		174 Tags	Redundancy
Social Tags	Moderate	Moderate	**Cheap**	Poor	Weak	Unstructured	Decent
Last.fm	Support Popular Website	Lots of Users	Website, Plugins, DB	Sparse in Long Tail		>960,000 Tags, $\sim$1000 Useful Tags	Adhoc Tagging
Game	Moderate	Moderate	**Cheap**	Decent	Depends on	Structured	**Good**
Listen Game	Design, Deploy, Promote Game	Trade Entertainment for Tags	Game Server, DB	Sparse in Long Tail	Game	174 Tags	Competition, Redundancy
Web Documents	**Cheap**	**None**	Moderate	Poor	Weak	Unstructured	Decent
RS System	Fully Automated		Webcrawling, Text Processing	Sparse in Long Tail		Natural Language	Noisy Channel
Autotags	**Cheap**	**Cheap**	Expensive	**Great**	**Strong**	Structured	Decent
SML Model	Fully Automated	Training Data Set	Audio Processing & Modeling	Label all Songs w/ all Tags		Depends on Training Data	Content-based analysis

Table 1. Approaches for Collecting Music Tags. The bold font indicates a strength for an approach.

Approach	Strengths	Weaknesses
Survey	custom-tailored vocabulary high-quality annotations strong labeling	small, predetermined vocabulary human-labor intensive time consuming approach lacks scalability
Social Tags	collective wisdom of crowds unlimited vocabulary provides social context	create & maintain popular social website ad-hoc annotation behavior, weak labeling sparse/missing in long-tail
Game	collective wisdom of crowds entertaining incentives produce high-quality annotations fast paced for rapid data collection	"gaming" the system difficult to create viral gaming experience listening to short-clips, rather than entire songs
Web Documents	large, publicly-available corpus of relevant documents no direct human involvement provides social context	noisy annotations due to text-mining sparse/missing in long-tail weak labeling
Autotags	not affected by cold-start problem no direct human involvement strong labeling	computationally intensive limited by training data based solely on audio content

Table 2. Strengths and weaknesses of tag-based music annotation approaches

suggest any free-text token to describe music. This vocabulary is also unstructured since tags are not organized in any way. In general, we prefer an extensible vocabulary because a fixed vocabulary limits text-based retrieval to a small set of predetermined tags. In addition, a structured vocabulary is advantageous since the ontological relationships (e.g., genre hierarchies, families of instruments) between tags encode valuable semantic information that is useful for retrieval.

Finally, the accuracy with which tags are applied to songs is perhaps the most important point of comparison. Since there is no ideal ground truth and listeners do not always agree whether (or to what degree) a tag should be applied to a song (i.e., 'the subjectivity problem' [15]), evaluating accuracy can be tricky. Intuitively, it is preferable to have trained musicologists, rather than untrained non-experts, annotate a music corpus. It is also advantageous to have multiple individuals, rather then a single person, annotate each song. Lastly, individuals who are given incentives to provide good annotations (e.g., a high score in a game) may provide better annotations than unmotivated individuals.

2.1 Conducting a Survey

Perhaps the most well-known example of the music annotation survey is Pandora's [1] "Music Genome Project" [5, 23]. Pandora uses a team of approximately 50 expert music reviewers (each with a degree in music and 200 hours of training) to annotate songs using structured vocabularies of between 150-500 'musically objective' tags depending on the genre of the music [8]. Tags, such as "Afro-Latin Roots", "Electric Piano Riffs" and "Political Lyrics", can be considered objective since, according to Pandora, there is a high level of inter-reviewer agreement when annotating the same song. Between 2000 and 2007, Pandora annotated over 600,000 songs [23]. Currently, each song takes between 20 to 30 minutes to annotate and approximately 15,000 new songs are annotated each month. While this labor-intensive approach results in high-quality annotations, Pandora must be very selective of which songs they choose to annotate given that there are already millions of songs by millions of artists [2].

[1] www.pandora.com

[2] In February 2008, Last.fm reported that their rapidly growing database consisted of 150 million songs by 16 million artists.

Pandora, as well as companies like Moodlogic [3] and All Media Guide (AMG) [4], have devoted considerable amounts of money, time and human resources to annotate their music databases with high-quality tags. As such, they are unlikely to share this data with the MIR research community. To remedy this problem, we have collected the CAL500 data set of annotated music [19]. This data set contains one song from 500 unique artists each of which have been manually annotated by a minimum of three non-expert reviewers using a structured vocabulary of 174 tags. While this is a small data set, it is strongly labeled, relies on multiple reviews per song, and as such, can be used as a standard data set for training and/or evaluating tag-based music retrieval systems.

2.2 Harvesting Social Tags

Last.fm [5] is a music discovery website that allows users to contribute *social* tags through a text box in their audio player interface. By the beginning of 2007, their large base of 20 million monthly users have built up an unstructured vocabulary of 960,000 free-text tags and used it to annotated millions of songs [16]. Unlike the Pandora and AMG, Last.fm makes much of this data available to the public through their Audiocrobbler [6] site. While this data is a useful resource for the MIR community, Lamere and Celma [11] point out a number of problems with social tags. First, there is often a sparsity of tags for new and obscure artists (cold start problem / popularity bias). Second, most tags are used to annotate artists rather than individual songs. This is problematic since we are interested in retrieving semantically relevant songs from eclectic artists. Third, individuals use ad-hoc techniques when annotating music. This is reflected by use of polysemous tags (e.g., "progressive"), tags that are misspelled or have multiple spellings (e.g., "hip hop", "hip-hop"), tags used for self-organization (e.g., "seen live"), and tags that are nonsensical. Finally, the public interface allows for malicious behavior. For example, any individual or group of individuals can annotate an artist with a misleading tag.

2.3 Playing Annotation Games

At the 2007 ISMIR conference, music annotation games were presented for the first time: ListenGame [20], Tag-a-Tune [12], and MajorMiner [14]. ListenGame is a real-time game where a large group of users is presented with a song and a list of tags. The players have to choose the best and worst tags for describing the song. When a large group of players agree on a tag, the song has a strong (positive or negative) association with the tag. This game, like a music survey, has the benefit of using a structured vocabulary of tags. It can be considered a strong labeling approach since it also collects information that reflects negative semantic associations between tags and songs. Like the ESPGame for image tagging [22], Tag-a-Tune is a two-player game where the players listen to a song and are asked to enter "free text" tags until they both enter the same tag. MajorMiner is similar in nature, except the tags entered by the player are compared against the database of previously collected tags in an offline manner. Like social tagging, the tags collected using both games result in a unstructured, extensible vocabulary.

A major problem with this game-based approach is that players will inevitably attempt to *game* the system. For example, the player may only contribute generic tags (e.g., "rock", "guitar") even if less common tags provide a better semantic description (e.g., "grunge", "distorted electric guitar"). Also, despite the recent academic interest in music annotation games, no game has achieved large scale success. This reflects the fact that it is difficult to design a viral game for this inherently laborious task.

2.4 Mining Web Documents

Artist biographies, album reviews, and song reviews are another rich source of semantic information about music. There are a number of research-based MIR systems that collect such documents from the Internet by querying search engines [9], monitoring MP3 blogs [4], or crawling a music site [24]. In all cases, Levy and Sandler point out that such web mined corpora can be *noisy* since some of the retrieved webpages will be irrelevant, and in addition, much of the text content on relevant webpages will be useless [13].

Most of the proposed web mining systems use a set of one or more documents associated with a song and convert them into a single document vector (e.g., tf-idf representation) [10, 25]. This *vector space* representation is then useful for a number of MIR tasks such as calculating music similarity [25] and indexing content for a text-based music retrieval system [10]. More recently, Knees et. al. [9] have proposed a promising new web mining technique called *relevance scoring* as an alternative to the vector space approaches. Both relevance scoring and vector space approaches are subject to popularity bias since short-head songs are generally represented by more documents than long-tail songs.

2.5 Autotagging Audio Content

All previously described approaches require that a song be annotated by humans, and as such, are subject to the cold start problem. Content-based audio analysis is an alternative approach that avoids this problem. Early work on this topic focused (and continues to focus) on music classification by genre, emotion, and instrumentation (e.g., [21]). 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 [19, 7, 17]. In [19], we describe a generative approach that learns a Gaussian mixture model (GMM) distribution over an audio feature space for each tag in the vocabulary. Eck et. al. use a discriminative approach by learning a boosted decision stump classifier for each tag [7]. Finally, Sordo et. al. present a non-parametric approach that uses a content-based measure

[3] http://en.wikipedia.org/wiki/MoodLogic
[4] www.allmusic.com
[5] www.last.fm
[6] http://www.audioscrobbler.net/

of music similarity to propagate tags from annotated songs to similar songs that have not been annotated [17].

3 COMPARING SOURCES OF TAGS

In this section, we describe one system for each of the tag collection approaches. Each has been implemented based on systems that have been recently developed within the MIR research community [19, 9, 20]. Each produces a $|S|$x$|T|$ *annotation matrix* $\mathbf{X}$ where $|S|$ is the number of songs in our corpus and $|T|$ is the size of our tag vocabulary. Each cell $x_{s,t}$ of the matrix is proportional to the strength of semantic association between song s and tag t.

We set $x_{s,t} = \emptyset$ if the relationship between song s and tag t is missing (i.e., unknown). If the matrix $\mathbf{X}$ has many empty cells, then we refer to the matrix as *sparse*, otherwise we refer to it as *dense*. Missing data results from both weak labeling and the cold start problem. Sparsity is reflected by the *tag density* of a matrix which is defined as the percentage of non-empty elements of a matrix.

Our goal is to find a tagging system that is able to accurately retrieve (i.e., rank-order) songs for a diverse set of tags (e.g., emotions, genres, instruments, usages). We quantitatively evaluate music retrieval performance of system a by comparing the matrix $\mathbf{X}^a$ against the CAL500 matrix $\mathbf{X}^{\text{CAL500}}$ (see Section 2.1). The $\mathbf{X}^{\text{CAL500}}$ matrix is a binary matrix where $x_{s,t} = 1$ if 80% of the individuals annotate song s with tag t, and 0 otherwise (see Section V.a of [19] for details). For the experiments reported in this section, we use a subset of 109 of the original 174 tags. [7] We will assume that the subset of 87 songs from the Magnatunes [6] collection that are included in the CAL500 data set are representative of long-tail music. As such, we can use this subset to gauge how the various tagging approaches are affected by popularity bias. [8]

Each system is compared to the CAL500 data set using a number of standard information retrieval (IR) evaluation metrics [9]: area under the receiver operation characteristic curve (AROC), average precision, R-precision, and Top-10 precision. An ROC curve is a plot of the true positive rate as a function of the false positive rate as we move down this ranked list of songs. The area under the ROC curve (AROC) is found by integrating the ROC curve and is upper-bounded by 1.0. A random ranking of songs will produce an expected AROC score of 0.5. Average precision is found by moving down our ranked list of test songs and averaging the precisions at every point where we correctly identify a relevant song. R-Precision is the precision of the top R-ranked songs where R is the total number of songs in the ground truth that have been annotated with a given tag. Top-10 precision is the precision after we have retrieved the top 10 songs for a given tag. This metric is designed to reflect the 10 items that would be displayed on the first results page of a standard Internet search engine.

Each value in Table 3 is the mean of a metric after averaging over all 109 tags in our vocabulary. That is, for each tag, we rank-order our 500 song data set and calculate the value of the metric using CAL500 data as our ground truth. We then compute the average of the metric using the 109 values from the 109 rankings.

3.1 Social Tags: Last.fm

For each of our 500 songs, we attempt to collect two lists of social tags from the Last.fm Audioscobbler website. One list is related specifically to the song and the other list is related to the artist. For the song list, each tag has a score ($x_{s,t}^{\text{Last.fm\_Song}}$) that ranges from 0 (low) to 100 (high) and is a secret function (i.e., trade secret of Last.fm) of both the number and diversity of users who have annotated song s with tag t. For the artist list, the tag score ($x_{s,t}^{\text{Last.fm\_Artist}}$) is again a secret function that ranges between 0 and 100, and reflects both tags that have been used to annotate the artist or songs by the artist. We found one or more tags for 393 and 472 of our songs and artists, respectively. This included at least one occurrence of 71 and 78 of the 109 tags in our vocabulary. While this suggests decent coverage, tag densities of 4.6% and 11.8%, respectively, indicate that the annotation matrices, $\mathrm{X}^{\text{Last.fm\_Song}}$ and $\mathrm{X}^{\text{Last.fm\_Artist}}$, are sparse even when we consider mostly short-head songs. These sparse matrices achieve AROC of 0.57 and 0.58.

To remedy this problem, we create a single Last.fm annotation matrix by leveraging the Last.fm data in three ways. First, we match tags to their synonyms. [9] For example, a song is considered to be annotated with 'down tempo' if it has instead been annotated with 'slow beat'. Second, we allow wildcard matches for each tag. That is, if a tag appears as a substring in another tag, we consider it to be a wildcard match. For example, "blues" matches with "delta electric blues", "blues blues blues", "rhythm & blues". Although synonyms and wildcard matches add noise, they increase the respective densities to 8.6% and 18.9% and AROC performance to 0.59 and 0.59. Third, we combine the song and artist annotation matrices in one annotation matrix:

$$\mathrm{X}^{\text{Last.fm}} = \mathrm{X}^{\text{Last.fm\_Song}} + \mathrm{X}^{\text{Last.fm\_Artist}}.$$

This results in a single annotation matrix that has a density of 23% and AROC of 0.62. 95 of the 109 tags are represented at least once in this matrix. However, the density for the Magnatunes (e.g., long-tail) songs is only 3% and produces retrieval results that are not much better than random.

3.2 Games: ListenGame

In [20], Turnbull et al. describe a music annotation game called ListenGame in which a community of players listen to a song and are presented with a set of tags. Each player is asked to vote for the single *best* tag and single *worst* tag to

[7] We have merged genre-best tags with genre tags, removed instrument-solo tags, removed some redundant emotion tags, and pruned other tags that are used to annotate less than 2% of the songs. For a complete list of tags, see http://cosmal.ucsd.edu/cal.

[8] It should be noted that 87 songs is a small sample.

[9] Synonyms are determined by the author using a thesaurus and by exploring the Last.fm tag vocabulary.

Approach	Songs	Tag Density	AROC	Avg. Prec	R-Prec	Top10 Prec
Survey (CAL500)	All Songs	1.00	1.00	1.00	1.00	0.97
Ground Truth	Long Tail	1.00	1.00	1.00	1.00	0.57
Baseline	All Songs	1.00	0.50	0.15	0.14	0.13
Random	Long Tail	1.00	0.50	0.18	0.15	0.12
Social Tags	All Songs	0.23	0.62	0.28	0.30	0.37
Last.fm	Long Tail	0.03	0.54	0.24	0.20	0.19
Game ListenGame†	All Songs	0.37	0.65	0.28	0.28	0.32
Web Documents	All Songs	0.67	0.66	0.29	0.29	0.37
SS-WRS	Long Tail	0.25	0.56	0.25	0.20	0.18
Autotags	All Songs	1.00	0.69	0.29	0.29	0.33
SML	Long Tail	1.00	0.70	0.34	0.30	0.27
Rank-based	All Songs	1.00	0.74	0.32	0.34	0.38
Interleaving (RBI)	Long Tail	1.00	0.71	0.33	0.27	0.28

Table 3. Tag-based music retrieval: Each approach is compared using all *CAL500* songs and a subset of 87 more obscure *long-tail* songs from the Magnatunes dataset. *Tag Density* represents the proportion of song-tag pairs that have a non-empty value. The four evaluation metrics (*AROC, Average Precision, R-Precision, Top-10 Precision*) are found by averaging over 109 tag queries. †Note that ListenGame is evaluated using half of the CAL500 songs and that the results do not reflect the realistic effect of the popularity bias (see Section 3.2).

describe the music. From the game, we obtain the annotation matrix $\mathbf{X}^{\mathbf{Game}}$ by letting

$$[\mathbf{X}^{\mathrm{Game}}]_{s,t} = \#(\text{best votes}) - \#(\text{worst votes})$$

when song s and tag t are presented to the players.

During a two-week pilot study, 16,500 annotations (best and worst votes) were collected for a random subset of 250 CAL500 songs. Each of the 27,250 song-tag pairs were presented to users an average of 1.8 times. Although this represents a very small sample size, the mean AROC for the subset of 250 songs averaged over the 109-tag vocabulary is 0.65. Long-tail and short-head results do not accurately reflect the real-world effect of popularity bias since all songs were selected for annotation with equal probability. As such, these results have been omitted.

3.3 Web Documents: Weight-based Relevance Scoring

In order to extract tags from a corpus of web documents, we adapt the relevance scoring (RS) algorithm that has recently been proposed by Knees et. al. [9]. They have shown this method to be superior to algorithms based on vector space representations. To generate tags for a set of songs, the RS works as follows:

1. **Collect Document Corpus:** For each song, repeatedly query a search engine with each song title, artist name, or album title. Collect web documents in search results. Retain the (many-to-many) mapping between songs and documents.
2. **Tag Songs:** For each tag
   (a) Use the tag as a query string to find the relevant documents, each with an associated *relevance weight* (defined below) from the corpus.
   (b) For each song, sum the relevance scores for all the documents that are related to the song.

We modify this algorithm in two ways. First, the relevance score in [9] is inversely proportional to the rank of the relevant document. We use a weight-based approach to relevance scoring (WRS). The relevance weight of a document given a tag can be a function of the number of times the tag appears in the document (tag-frequency), the number of documents with the tag (document frequency), the number of total words in the document, the number of words or documents in the corpus, etc. For our system, the relevance weights are determined by the MySQL match function. [10]

We calculate an entry of the annotation matrix $\mathbf{X}^{\mathrm{WRS}}$ as,

$$\mathbf{X}^{\mathrm{WRS}}_{s,t} = \sum_{d \in D_t} w_{d,t} I_{d,s}$$

where D_t is the set of relevant documents for tag t, $w_{d,t}$ is the relevance weight for document d and tag t, and $I_{d,s}$ is an indicator variable that is 1 if document d was found when querying the search engine with song s (in Step 1) and 0 otherwise. We find that weight-based RS (WRS) produces a small increase in performance over rank-based RS (RRS) (AROC of 0.66 vs. 0.65). In addition, we believe that WRS will scale better since the relevance weights are independent of the number of documents in our corpus.

The second modification is that we use *site-specific* queries when creating our corpus of web documents (Step 1). That is, Knees et. al. collect the top 100 documents returned by Google when given queries of the form:

- "<artist name>" music
- "<artist name>" "<album name>" music review
- "<artist name> " "<song name>" music review

for each song in the data set. Based on an informal study of the top 100 webpages returned by non-site-specific queries, we find that many pages contain information that is only

[10] http://dev.mysql.com/doc/refman/5.0/en/fulltext-natural-language.html

slightly relevant (e.g., music commerce site, ticket resellers, noisy discussion boards, generic biographical information). By searching music-specific sites, we are more likely to find detailed music reviews and in-depth artist biographies. In addition, the webpages at sites like Pandora and AMG All Music specifically contain useful tags in addition to natural language content.

We use site-specific queries by appending the substring 'site:<music site url>' to the three query templates, where <music site url> is the url for a music website that is known to have high quality information about songs, albums or artists. These sites include allmusic.com, amazon.com, bbc.co.uk, billboard.com, epinions.com, musicomh.com, pandora.com, pitchforkmedia.com, rollingstone.com, wikipedia.org. For these 10 music sites and one non-site-specific query, we collect and store the top 10 pages returned by the Google search engine. This results in a maximum of 33 queries and a maximum of 330 pages per song. On average, we are only able to collect 150 webpages per song since some of the long-tail songs are not well represented by these music sites.

Our *site-specific weight-based relevance scoring* (SSWRS) approach produces a relatively dense annotation matrix (46%) compared with the approach involving Last.fm tags. However, like the Last.fm approach, the density of the annotation matrix is greatly reduced (25%) when we consider only long-tail songs.

3.4 Autotagging: Supervised Multiclass Labeling

In [19], we use a supervised multiclass labeling (SML) model to automatically annotate songs with a diverse set of tags based on audio content analysis. The SML model is parameterized by one Gaussian mixture model (GMM) distribution over an audio feature space for each tag in the vocabulary. The parameters for the set of GMMs are trained using annotated training data. Given a novel audio track, audio features are extracted and their likelihood is evaluated using each of the GMMs. The result is a vector of probabilities that, when normalized, can be interpreted as the parameters of a multinomial distribution over the tag vocabulary. This *semantic multinomial* distribution represents a compact and interpretable index for a song where the large parameter values correspond to the most likely tags.

Using 10-fold cross validation, we can estimate a semantic multinomial for each of the CAL500 songs. By stacking the 50 test set multinomials from each of the 10 folds, we can construct a strongly-labeled annotation matrix $\mathbf{X}^{\text{SML}}$ that is based purely on the audio content. As such, this annotation matrix is dense and not affected by the cold start problem.

3.5 Summary

Comparing systems using a two-tailed, paired t-test (N = 109, $\alpha = 0.05$) on the AROC metric, we find that all pairs of the four systems are significantly different, with the exception of Game and Web Documents. [11] If we compare the systems using the other three metrics (Average Precision, R-Precision, and Top 10 Precision), we no longer find statistically significant differences. It is interesting that Social Tags and Web Documents (0.37) have slightly better Top 10 precision than Autotags (0.33). This reflects the fact that for some of the more common individual tags, we find that Social Tags and Web Documents have exceptional precision at low recall levels. For both Web Documents and Social Tags, we find significant improvement in retrieval performance of short-head songs over long-tail songs. However, as expected, there is no difference for Autotags. This confirms the intuition that systems based on web documents and social tags are influenced by popularity bias, whereas content-based autotagging systems are not.

4 COMBINING SOURCES OF TAGS

While the purpose of this paper is to compare various approaches for collecting tags for music, our ultimate goal is to combine these approaches in order to create a more powerful tag-based music retrieval system. For example, if we interleave the top ranked songs from each of the four approaches (See Rank-based Interleaving (RBI) in Table 3), we observe a significant increase in performance (AROC 0.74) over the performance of the best single approach (Autotags with AROC = 0.69). Our improvement is consistent with the findings of Yoshii et. al. [26] and Aucountier et. al. [1], both of whom have recently proposed hybrid context-content systems for music recommendation and music classification, respectively. We explore alternative hybrid systems in some of our related work [2, 18].

5 ACKNOWLEDGEMENTS

We would like to thank our anonymous reviewers who had a large impact on this paper. This work is supported by NSF IGERT DGE-0333451 and NSF grant DMS-MSPA 062540922.

6 REFERENCES

[1] J.J. Aucouturier, F. Pachet, P. Roy, and A. Beurive. Signal + context = better classification. *ISMIR*, 2007.

[2] L. Barrington, M. Yazdani, D. Turnbull, and G. Lanckriet. Combining feature kernels for semantic music retrieval. *ISMIR*, 2008.

[3] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE PAMI*, 29(3):394–410, 2007.

[4] O. Celma, P. Cano, and P. Herrera. Search sounds: An audio crawler focused on weblogs. In *ISMIR*, 2006.

[5] S. Clifford. Pandora's long strange trip. *Inc.com*, 2007.

[6] J. S. Downie. Music information retrieval evaluation exchange (MIREX), 2005.

[7] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Neural Information Processing Systems Conference (NIPS)*, 2007.

[8] W. Glaser, T. Westergren, J. Stearns, and J. Kraft. Consumer item matching method and system. *US Patent Number 7003515*, 2006.

[11] Note that when we compare each system with the Game system, we compare both systems using the reduced set of 250 songs.

[9] P. Knees, T. Pohle, M. Schedl, D. Schnitzer, and K. Seyerlehner. A document-centered approach to a natural language music search engine. In *ECIR*, 2008.

[10] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A music search engine built upon audio-based and web-based similarity measures. In *ACM SIGIR*, 2007.

[11] P. Lamere and O. Celma. Music recommendation tutorial notes. ISMIR Tutorial, September 2007.

[12] E. L. M. Law, L. von Ahn, and R. Dannenberg. Tagatune: a game for music and sound annotation. In *ISMIR*, 2007.

[13] M. Levy and M. Sandler. A semantic space for music derived from social tags. In *ISMIR*, 2007.

[14] M. Mandel and D. Ellis. A web-based game for collecting music metadata. In *ISMIR*, 2007.

[15] C. McKay and I. Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? *ISMIR*, 2006.

[16] F. Miller, M. Stiksel, and R. Jones. Last.fm in numbers. *Last.fm press material*, February 2008.

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

[18] D. Turnbull. *Design and Development of a Semantic Music Discovery Engine*. PhD thesis, UC San Diego, 2008.

[19] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE TASLP*, 16(2), 2008.

[20] D. Turnbull, R. Liu, L. Barrington, and G Lanckriet. Using games to collect semantic information about music. In *ISMIR '07*, 2007.

[21] 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.

[22] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *ACM CHI*, 2004.

[23] T. Westergren. Personal notes from Pandora get-together in San Diego, March 2007.

[24] B. Whitman and D. Ellis. Automatic record reviews. *ISMIR*, pages 470–477, 2004.

[25] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. *ICMC*, 2002.

[26] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE TASLP*, 2008.