Correction

COMPUTER SCIENCES

The authors note that, due to a printer’s error, the affiliation for Luke Barrington and Gert Lanckriet should instead appear as “Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92093.” The corrected author and affiliation lines appear below. The online version has been corrected.

Luke Barrington, Douglas Turnbull, and Gert Lanckriet

*Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92093; and Computer Science Department, Ithaca College, Ithaca, NY 14850

www.pnas.org/cgi/doi/10.1073/pnas.1205806109
Game-powered machine learning

Luke Barrington*1, Douglas Turnbull*2, and Gert Lanckriet*1

*Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92093; and *Computer Science Department, Ithaca College, Ithaca, NY 14850

Edited* by Grace Wahba, University of Wisconsin-Madison, Madison, WI, and approved December 27, 2011 (received for review October 6, 2010)

Searching for relevant content in a massive amount of multimedia information is facilitated by accurately annotating each image, video, or song with a large number of relevant semantic keywords, or tags. We introduce game-powered machine learning, an integrated approach to annotating multimedia content that combines the effectiveness of human computation, through online games, with the scalability of machine learning. We investigate this framework for labeling music. First, a socially-oriented music annotation game called Herd It collects reliable music annotations based on the “wisdom of the crowds.” Second, these annotated examples are used to train a supervised machine learning system. Third, the machine learning system actively directs the annotation games to collect new data that will most benefit future model iterations. Once trained, the system can automatically annotate a corpus of music much larger than what could be labeled using human computation alone. Automatically annotated songs can be retrieved based on their semantic relevance to text-based queries (e.g., “funk jazz with saxophone,” “spooky electronic,” etc.).

To enable semantic search of nontextual content requires a mapping between multimedia data and a wide vocabulary of descriptive tags. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).

To make semantic search of nontextual content possible, a mapping between multimedia data and a wide vocabulary of descriptive tags is required. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).

To enable semantic search of nontextual content requires a mapping between multimedia data and a wide vocabulary of descriptive tags. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).

To enable semantic search of nontextual content requires a mapping between multimedia data and a wide vocabulary of descriptive tags. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).

To enable semantic search of nontextual content requires a mapping between multimedia data and a wide vocabulary of descriptive tags. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).

To enable semantic search of nontextual content requires a mapping between multimedia data and a wide vocabulary of descriptive tags. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).

To enable semantic search of nontextual content requires a mapping between multimedia data and a wide vocabulary of descriptive tags. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).

To enable semantic search of nontextual content requires a mapping between multimedia data and a wide vocabulary of descriptive tags. Describing multimedia content with relevant semantics necessitates intervention from humans who can understand and interpret the images, video, or music. However, manual tagging by human experts is too costly and time-consuming to be applied to billions of data items. For example, Pandora, a popular music retailer, offers a growing catalog of more than 20 million songs. Developing a semantic multimedia search engine—that enables simple discovery of relevant multimedia content as easily as Internet search engines [e.g., Google (4)]—helps find relevant web pages—presents a challenge because the domain of the query (text) differs from the range of the search results (images, video, music).
We deploy this game-based machine learning system to investigate and answer two important questions. First, we demonstrate that the collective wisdom of Herd It’s crowd of nonexperts can train machine learning algorithms as well as expert annotations by paid musicologists. In addition, our approach offers distinct advantages over training based on static expert annotations: it is cost-effective, scalable, and has the flexibility to model demographic and temporal changes in the semantics of music. Second, we show that integrating Herd It in an active learning loop trains accurate tag models more effectively; i.e., with less human effort, compared to a passive approach.

Herd It—a Social Music Annotation Game
A player arriving at Herd It (www.HerdIt.org) is connected with “the Herd”—all other players currently online—and the game begins. Each round of Herd It begins by playing the same piece of music to all members of the Herd. A variety of fun, simple mini-games prompt players to choose from suggested tags that describe different aspects of the music they hear (Fig. 2 illustrates an example of Herd It’s gameplay with further examples in SI Text). In every minigame, players earn points based on their agreement with the tags chosen by the rest of the Herd, encouraging players to contribute tags that are likely to achieve consensus.

Herd It’s goal is to collect training data that primes and improves the machine learning system through an active learning loop by motivating human players to provide reliable descriptions of a large number of example songs using a dynamic vocabulary of tags. To achieve this goal, Herd It’s development followed a user-centered design process (16) that aimed to create an intuitive, viral game experience. A series of rapid prototypes were released every month and tested on focus groups of 5–50 new players, both in person at our lab and in a controlled online environment. During each test, we evaluated factors including playability and appeal, user-interface intuitiveness, viral potential, and stability. Interviews and questionnaires were used to evaluate the extent to which players were able to focus on the music (ensuring reliable data collection), their awareness of the other players (Herd It is a social game and a player’s score depends on the Herd), and overall enjoyment (indicating likelihood of large-scale participation). Iterative user feedback led to improvements in the design and the process continued until key gameplay and social evaluation metrics were satisfied (e.g., 94% of players understood the scoring metric within five games, 82% said they would recommend Herd It to their friends; further results in SI Text). The user-centered design process was instrumental in determining crucial gameplay mechanics, described below, that differentiate Herd It from other music annotation games [e.g., (13, 15)].

In particular our user tests discovered that, while free-text tagging works well when annotating images (which tend to feature many obvious, easily named objects; see; e.g., ref. 11), many listeners found it difficult to produce and agree on a variety of tags for music in a game environment without some priming. Asking players to type their own descriptions of the music meant that the vast majority of tags were confined to a limited vocabulary of generic tags; e.g., “rock,” “guitar,” “drums,” “male/female vocalist” [MajorMiner (15) suffers from this problem]. As a result,
the independent inputs of multiple players rarely converged on more interesting tags [TagATune (13) avoids this problem by asking players to guess whether they are listening to the same song, based on the free-text tags other players entered, rather than requiring agreement on the exact tags]. To achieve both variety and consensus, Herd It’s unique solution is to suggest tags for player confirmation, thereby controlling the vocabulary used to describe music while maintaining simple and compelling gameplay. In addition, tag suggestion addresses another, important design objective: it facilitates the active learning paradigm depicted in Fig. 1 which requires precise control over the data collected from the Herd. Specifically, an active learning approach leverages machine learning models to suggest \(\text{song,tag}\) combinations that, if confirmed by human players, are most likely to produce useful training examples and optimize future model training. Herd It’s tag suggestion mechanism enables active learning by focusing human labeling on specific \(\text{song,tag}\) combinations. Vice versa, Herd It’s new tag suggestion design benefits from it being powered with machine intelligence. Indeed, suggesting tags randomly, rather than intelligently, would produce too many minigames that have no relevant choices and are not fun.

To achieve widespread player engagement and thus maximize training data collection, we found that Herd It should target the “casual” gamer. Unlike the traditional computer gaming demographic (i.e., teenage boys who enjoy long-lasting games with complicated gameplay mechanics, casual games appeal to a much wider demographic (e.g., skewed towards middle-aged women), are played in short time increments (5–20 min) and feature simple but addictive gameplay (17). Herd It’s simple, single-click gameplay, cartoon-ish minigame design, and intuitive scoring metric were designed to attract a broad audience of casual gamers. Based on the choices offered in a given minigame, different users may end up describing a song differently, using either compatible tags (e.g., a “romantic” song that is also described as “carefree”) or opposite tags (e.g., what sounds “exciting” to one listener may be “boring” to another). Given this subjectivity inherent in music appreciation, our design process revealed that it is important to evaluate agreement in minigames in a (larger) group setting, as this enables clusters of consensus to develop between the players, around multiple “right” answers. This observation inspired us to make “the Herd” a central feature of the game, rather than the player-vs-player mechanic used by other games (e.g., (11, 13)). In addition, our user tests determined that real-time, social interaction produced more compelling gameplay than off-line group feedback [e.g., (15)]. The group dynamic also makes it more difficult for a few players to cheat and gain lots of points by coordinating poor labeling (other measures to prevent cheating include randomizing tag order in minigames and preventing a single player from entering multiple games).

Finally, because individuals use music preference to communicate information about their personality (18), players desired Herd It to be embedded in a larger social music experience. For example, players wanted to choose preferred genres, share music, create personal profiles, and challenge and compare scores with friends, leading us to integrate the game within the players’ existing social network by releasing Herd It as an application on Facebook. Integrating Herd It with Facebook offers many avenues to engage players (e.g., easy login, personalized messages, player photos) and promote the game to a wide audience (e.g., invites, challenges, see ref. 23). Facebook also provides demographic and psychographic information about players (e.g., gender, age, location, friends, favorite music), offering a hitherto unavailable level of insight into how different people experience and describe music.

### Automatic Music Tagging

Statistical pattern recognition methods for tagging music begin by extracting features that summarize properties of the acoustic waveforms, essentially “listening” to the musical signal. By considering a training set of reliably labeled songs, supervised machine learning algorithms identify statistical regularities in these acoustic features that are predictive of descriptive tags like “bluegrass,” “banjo,” or “mellow.” Machines can then generalize this knowledge by detecting the presence of similar patterns in vast catalogs of new, untagged music, thereby leveraging the accuracy of human labeling (to obtain the training set) with the scalability of automated analysis to tag this new music content. Machine learning methods for music tagging continue to improve and, given training data of sufficient quality, their accuracy approaches the ceiling set by the inherent subjectivity in describing music with tags (19).

To thoroughly evaluate the efficacy of the game-powered machine learning paradigm depicted in Fig. 1, we consider various state-of-the-art autotagging algorithms for its machine learning component, including generative (19, 20) and discriminative (21, 22) approaches. Generative methods focus on estimating the (class-conditional) distribution \([\text{e.g., with a Gaussian mixture model (GMM), dynamic texture mixtures (DTM), etc.}]\) of acoustic features that are common among songs that human “trainers” have labeled with a given tag. By evaluating the likelihood of features from a new song under the learned distribution, the model determines the probability that the tag is a relevant description of the song (19). Discriminative methods, on the other hand, directly optimize a decision rule to discriminate between a tag being present or absent for a given audio clip. Evaluating the decision rule for a new song allows to obtain tag probabilities. Just as Internet search engines rank web pages by their relevance to a text query, the tag probabilities output by a model can be used to rank songs by their relevance to the tag.

Traditional machine learning approaches use a single, fixed training set to learn models that, once trained, remain static. In our game-powered machine learning framework however, new data is constantly being contributed by Herd It players. That data can be used to update our tag models. Even more, because Herd It’s design permits actively focusing players’ efforts on specific songs and tags, it is possible to collect specifically that data that is expected to improve tag models most effectively, achieved through an active learning approach (see ref. 23 for a review), that leverages the current tag models to identify the most effective song-tag pairs for future model updates. Active learning fully integrates the autotagging algorithm with the data collection process, to optimize model training. To investigate the benefits of deploying Herd It in an active learning loop compared to updating with randomly collected data, we develop a unique active learning algorithm to suggest data for training the generative GMM-based autotagger, a top performer in the 2008 MIREX evaluation of automatic music tagging algorithms (24).

Various active learning algorithms have been proposed for discriminative machine learning methods\(^\dagger\), where both positive and negative examples are used to learn a decision boundary between classes. Generative approaches, on the other hand, require only positively labeled examples for training (i.e., songs that exemplify a certain tag) and negatively labeled training examples offer no improvement to the model\(^\ddagger\). To collect positively labeled training examples and improve a generative model through active learning may suggest sampling unlabeled examples that have high likelihood under the current model and procuring labels for them [e.g., by presenting \(\text{song,tag}\) pairs in Herd It minigames]. However, this “certainty” sampling approach suffers from two drawbacks: early in training, when the model is not yet well learned,\(^\dagger\)

\(^\dagger\)Strategies for actively learning discriminative models include uncertainty sampling (25) where points are chosen that are least certain (or have highest entropy) under the current model (e.g., points closest to the decision boundary) and variance reduction (26) where samples are chosen to reduce the model’s output variance.

\(^\ddagger\)For example, for generative models, uncertainty sampling faces the problem that unlabeled songs which have low certainty under the current model are likely to result in negative labels.
the most likely samples may not in fact be positive examples and thus will not contribute to the training set. Later in the learning process, sampling from the most likely areas results in many confirmed positive examples that conform to the model’s current training set and lack the diversity required to generalize the current model to uncertain areas of the feature space. Exploration of these uncertain areas advocates for a more random sampling of unlabeled examples. Rather than a complete random sampling, we can actively increase the efficiency of the data collection and, thus, the learning rate, by reducing the likelihood of sampling unlabeled examples that are eventually labeled as negatives (which are of no use to train the generative model) and thereby avoiding points that most disagree with the current model. More specifically, we rank all of the unlabeled examples by their likelihood under the current model, remove the 10% of examples with lowest likelihood, and query labels randomly from the remaining 90% of examples. By removing the least likely points and sampling randomly elsewhere, we aim to avoid querying labels for negative examples and achieve rapid confirmation of a diverse training set for our generative model. Our active GMM experiments show that removing the 10% least likely songs finds a good balance between exploring poorly modeled areas of the feature space while avoiding points that are unlikely to produce positive examples.8

Game-Powered Machine Learning

Our game-powered machine learning approach aims to collect sufficient human labels, through game play, to train an automatic music annotation system that can reliably generalize semantics to unlimited music. Qualitatively, we argue that this crowdsourced approach is superior to requiring expert annotators as it is less costly, more scalable, and collects a dynamic dataset that can be adapted over time to focus on the most relevant or important topics. To quantify the efficacy of our game-powered machine learning framework, we conduct experiments designed to answer the following two questions: (i) Can machine learning algorithms be trained with data collected from Herd It’s crowd of nonexperts as accurately as with data collected from paid expert musicologists? (ii) Can accurate tag models be learned with less human effort by encapsulating Herd It in an active learning framework?

To answer these questions, we deployed Herd It online, engaging 7,947 people to provide over 140,000 clicks that associate songs with tags through five different types of minigames.

To generate minigames in a passive system, without active learning, we begin with 10–20 candidate {song,tag} pairs, chosen randomly from the authors’ personal music collection of over 6,000 popular songs from the past 70 y, and a vocabulary of 1,269 tags, including subgenres, emotions, instruments, usages, colors, and more categories. To generate a minigame, one {song,tag} pair is selected from the list of candidate pairs, biased by associations (determined using the online music service http://last.fm/api/) with the musical genre selected by the player at the start of the game (pop, rock, hip-hop, blues, electronics, or “everything”) and by the particular minigame (e.g., some minigames focus on subgenres, colors, or bipolar adjectives). Remaining minigame tags (each minigame suggests between one and nine tags) are restricted to the same tag category. Candidate {song,tag} pairs remain on the list until they have been viewed by at most 50 players. At that point the {song,tag} pair is discarded and replaced by a new, randomly sampled one. Maintaining a reasonable list of candidate pairs ensures diverse gameplay.

Consensus between players’ clicks collected in Herd It minigames is used to “confirm” reliable {song,tag} associations. More specifically, the generative model of labels, accuracies, and difficulties, or “GLAD,” (10) conceives of each human input as an estimate of the underlying true label that has been corrupted by player inaccuracy and the difficulty of labeling the song. Using an expectation-maximization algorithm, GLAD optimally combines the votes from all Herd It players and we confirm the findings of (10) that the resulting consensus is more reliable than heuristics such as majority vote, percentage agreement, or vote thresholds. A {song,tag} pair is presented in Herd It minigames until GLAD “confirms” a reliable association, based on the historical click data for that {song,tag} pair. If a {song,tag} pair remains unconfirmed after being viewed by 50 Herd It players, it is “rejected” and not sampled further. Overall, GLAD confirmed 8,784 {song,tag} pairs, representing song examples of 549 tags, while 256,000 pairs were rejected. To ensure that we have enough data to train robust machine learning models and answer the first question, we reduce the dataset to the 127 tags for which Herd It has identified at least 10 reliable example songs. Data was collected passively (i.e., no active learning) and this provides the baseline against which to compare an active learning strategy and evaluate the second question.

To answer the first question, we quantify the efficacy of our game-powered machine learning framework and compare it to “expert-trained” machine learning. That is, we evaluate the performance of a music autotagging algorithm when trained on (i) the Herd It game data and (ii) data derived from expert musicologists at Pandora.com’s “Music Genome Project” (MGP), respectively. After training, the accuracy of each autotagger is evaluated on CAL500: an independent evaluation set of 500 songs fully labeled by multiple humans using a controlled survey (19) (see SI Text for details about the CAL500 and MGP datasets). For this comparison, we train and evaluate models of all tags that are available in both the MGP and CAL500 vocabulary and for which Herd It has collected at least ten confirmed example songs, resulting in 25 tags. The models of each of these tags are used to retrieve the 10 most relevant songs from the CAL500 corpus for each single-tag query. These top-ten search results—automatically retrieved by a machine—are evaluated by comparing to the CAL500 “ground-truth,” and computing the precision (i.e., the number of songs in the machine-ranked top-ten that the ground truth effectively associates with the tag). Finally, the precision is averaged over all 25 tags. Because both Herd It and MGP models are instances of the same machine learning algorithm, but trained on different datasets, any significant differences in autotagging performance most likely reflect differences in the quality of the respective training data sources and allow us to evaluate human computation games—Herd It, in particular—as a source of reliable training data. To prevent bias induced by a particular choice of machine learning algorithm, this comparison is repeated for multiple state-of-the-art autotagging algorithms.

In addition to showing that game-powered machine learning can be competitive with an expert-trained system, in a second step, we demonstrate the efficacy of actively integrating machine learning with game-based data collection. The baseline here is the passive approach outlined above, which “analyzes” (i.e., confirms or rejects through human labeling) {song,tag} pairs in random order. As more {song,tag} pairs are analyzed (i.e., more human effort contributed), a tag’s training set grows, tag models are updated, and autotagging performance is expected to improve. We compare this passive approach to an active learning paradigm which aims to improve tag models more effectively by leveraging current models to select the next {song,tag} pairs that will be analyzed. More precisely, for each of the 25 Herd It tags that were evaluated earlier, we collect all songs that appeared with the tag (confirmed or rejected) in previous Herd It minigames. We then estimate 25 GMM-based tag models by engaging in an iterative training procedure, for each tag, based on this list of “candidate” songs. At each iteration, we first compute the likelihood, under the current tag model, of all remaining candidate songs and use our active learning method for
generative models to prioritize 10 candidate songs for analysis with that tag (for the first iteration, candidate songs are chosen randomly). That is, we use our active learning algorithm to resample \{(song,tag)\} pairs that were previously presented in Herd It games. Songs for which the \{(song,tag)\} pair was previously confirmed are added to the tag’s training set; the remaining, rejected songs are removed from future candidate lists. Finally, we retrain the tag model using the updated training set and evaluate its performance on the CAL500 test set. We once again recompute the likelihood of all remaining candidate songs under the updated model, actively select 10 candidate songs for analysis, retrain the tag models, and so on. Song selection is repeated up to 200 times, analyzing up to 2,000 songs for each tag. At each iteration, we evaluate the average performance of the 25 tag models. We compare active learning to the passive baseline which samples 10 songs randomly at each iteration, for each tag.

Results
Table 1 presents the average precision of the top-ten music search results for 25 single-tag queries on CAL500, achieved by training four state-of-the-art autotagging algorithms on Herd It’s data. We compare to the performance obtained by training on expert MGP data. For each of the 25 tags common to Herd It, MGP and CAL500, we evaluate the top-ten precision on CAL500 and average performance over all tags. While the absolute performance depends on the machine learning method used, the relative performance between models trained using Herd It and those that use MGP data remains consistently over 95%. These findings answer our first question by demonstrating that a game-based machine learning system, trained on data collected from Herd It players, provides a competitive alternative to a system trained on expert labeled data, across a variety of algorithms.

Fig. 3 offers a more detailed comparison of Herd It and MGP based systems, by examining the performance of each tag model learned by the hierarchical GMM algorithm (19). The ability of the machine learning algorithm to model different tags varies; e.g., “acoustic,” “male lead vocals,” and “hip hop” songs are more easily identified, while “hand drums” and “funk” music are poorly modeled. In general, model performance is independent of the training data source (i.e., most points lie close to the diagonal in Fig. 3, indicating comparable results for each system). Models trained on either data source performed significantly differently for just two tags: “synthesizer” (MGP-based model better) and “drum set” (Herd It-based model better, 2-tailed t-test, 95% significance level). In summary, Table 1 and Fig. 3 demonstrate that training from Herd It’s crowdsourced data captures knowledge similar to training from expert annotations.

We turn now to the second question: can integrating machine learning and Herd It’s game-powered data collection in an active learning loop train accurate models with less human effort than a passive system? To measure human effort, we consider the number of \{(song,tag)\} pairs analyzed through Herd It gameplay, expressed as the number of songs analyzed per tag (for each of the 25 tags being modeled). Fig. 4 displays the improvement in song retrieval performance of the GMM autotagging algorithm as more songs are analyzed for each tag (and, consequently, more training examples collected for model estimation), following both an active learning and a random sampling strategy. The results demonstrate an improved learning rate due to active learning: active learning requires analyzing, on average, 450 songs per tag to achieve no significant difference between Herd It and MGP performance (paired, one-tailed t-test, \(p = 0.1\)) while the passive strategy hits this level after analyzing 940 songs for each tag. Fig. 4 highlights the improved efficiency by shading the learning curves while performance is significantly different from the MGP: by prioritizing the order in which \{(song,tag)\} pairs are presented to players, our active learning approach reduces the human labeling effort required by half. A more detailed inspection of the results reveals that active learning achieves expert performance by confirming an average of 31 training songs per tag, out of 450 analyzed candidates, vs. 49 out of 940 for random sampling. Active learning boosts the learning rate by suggesting fewer \{(song, tag)\} pairs that are eventually rejected and not used for training [compared to suggesting random \{(song,tag)\} pairs] while still producing a sufficiently diverse set of confirmed training songs.

Having demonstrated that Herd It data can train automatic music taggers that are as accurate as an expert-trained system, we now compare the tagging efficiency of crowdsourced amateur players with that of trained experts. Pandora’s musicological experts take 20–30 min to analyze and quantify the association between a song and 100–500 semantic dimensions, a rate of about 12 song-tag associations per expert-minute. Herd It minigames last about 30 s and present, on average, 5–4 tags for player analysis. Thus a single Herd It player analyzes 10.8 song-tag associations per minute, a little less than the Pandora expert. To quantify (i.e., confirm or reject) a song-tag association, the analysis of up to 50 players is required, vs. that of one Pandora expert. So, Herd It’s game-based approach gathers reliable tags from humans for free with about 2% the efficiency of paid, expert labeling. Of course, Herd It’s lower efficiency is multiplied by the number of simultaneous players in the Herd, which could be significantly larger than the number of musicological experts that can be gainfully employed, simultaneously.

In comparing game-based and expert annotation methods, we recognize that, even with crowdsourced consensus, multiple

![Table 1. Average top-ten precision of autotagging algorithms trained on Herd It examples and tested on CAL500](image-url)
amateur raters are likely less reliable than experts when identifying certain details pertaining to musical theory (e.g., we find Herd It players are inconsistent in Scales minigames that ask to distinguish major from minor keys) or esoteric subgenres and instruments (e.g., tags for the subgenres “dooom metal” and “worldbeat” and the instruments “siren” and “sponos” were rarely chosen when suggested in a minigame and, if they were chosen, it often was seemingly without relation to the audio content). In designing Herd It’s games, we generally focused on tags that are more relevant to our goal of building a music search engine that can empower a wide audience to discover relevant music using simple, semantic search.

Finally, while a human-only approach requires the same labeling effort for the first song as for the millionth, our game-powered machine learning solution needs only a small, reliable training set before all future examples can be labeled automatically, improving efficiency and cost by orders of magnitude. Tagging a new song takes 4 s on a modern CPU: in just a week, eight parallel processors could tag 1 million songs or annotate Pandora’s complete song collection, which required a decade of effort from dozens of trained musicologists.

Conclusions

We proposed game-powered machine learning as an integrated, scalable, affordable, and reliable solution for semantic search of massive amounts of multimedia content and investigated its efficacy for music search. Herd It, an online music annotation game, collects reliable examples of how humans use semantic tags to describe music. By itself, this human computation approach is insufficient to label the millions of songs available on the web. Instead, the knowledge collected by our game trains machine learning algorithms that can generalize tags to vast amounts of new, unlabeled music. Compared to other music games with a purpose, Herd It was specifically designed to be actively integrated with the machine learning algorithms and provide the data that most effectively trains them. Our results demonstrate, first, that game-powered machine learning is as good as expert-based machine learning—annotations collected from human computation games train autotagging models as accurately as expensive, expert annotations—while offering some distinct advantages (e.g., cost-effectiveness, scalability, flexibility to update the game to focus on tags of interest). Second, we show that embedding Herd It in an active learning paradigm trains accurate autotaggers more effectively; i.e., with less human effort, compared to a passive approach. We conclude that actively integrating human computation games and machine learning—combining targeted data collection by annotation games with automatic prediction by scalable machine learning algorithms—enables simple, widespread multimedia search and discovery.

ACKNOWLEDGMENTS. The authors thank Damien O’Malley for assistance in the design, development, and testing of Herd It, and Sanjoy Dasgupta and Brian McPhee for active learning advice. L.B. and D.T. received funding from a National Science Foundation (NSF) fellowship (DGE-0333451). L.B. received funding from the Qualcomm Innovation Fellowship. L.B. and G.L. received funding from the Hellman Fellowship Program, the von Liebig Center, the Committee on Research (grant RJ138G-LANCKRIET), the Alfred P. Sloan Foundation, Yahoo! Inc., and the NSF (DMS-MSPA 0625409 and IIS-1054960).

**Supporting Information**

Barrington et al. 10.1073/pnas.1014748109

**SI Text**

**Related Work on Human Computation**

Herd It and similar “games with a purpose” (1–6) offer fun and competition as incentives to motivate widespread human participation in scientific endeavors. Other human computation approaches engage participation on a volunteer basis (7–9) or use Amazon’s Mechanical Turk* to offer small monetary rewards in return for completing data labeling tasks (10–16). The majority of applications have focused on classifying text (8, 10, 13) or images (9, 11, 12, 15–17) although speech transcription (5, 14) and video labeling (7) applications also exist.

Beyond labeling of multimedia data, human computation methods have been applied to numerous fields where the so-called “wisdom of the crowds” provides insight beyond what individual experts can offer. Crowdsourcing successes include prediction markets for sports betting (18), product development scheduling (19), company stocks (20), and political races (21).

Data collected by annotation games have been used to evaluate the output of machine learning systems. E.g., data from the ESP-game (1) has been used as a computer vision test set (22) and both MajorMiner (2) and TagATune (4) have been used to evaluate and compare different music tagging algorithms (26, 27). To date, attempts to use human computation to train machine learning systems as accurately as training them from expert data have focused on using Amazon’s Mechanical Turk, rather than games. Novotney, et al. (14) use Mechanical Turk to crowdsource transcriptions of phone conversations and find a small reduction in performance of the resulting speech recognition system, compared to the same system trained on expert transcriptions. Ambati, et al. (13) also use Mechanical Turk to collect 3,000 English translations of Spanish sentences and train a machine translation system that rivals a system trained on expert annotations.

The proposed game-powered machine learning moves beyond monetary incentives and collects training data for free, a potentially more sustainable and scalable approach. For example, human computation games for annotating multimedia data have succeeded in collecting hundreds of thousands of tags for images (1) and music (4), a significantly larger scale than most Mechanical Turk applications [e.g., thousands to tens of thousands of tags for images (11) or speech transcription (14)]. We investigate whether online annotation games, which entice players with fun and real-time social interaction with a variety of other players, can train systems competitive with those trained from experts. We focus on labeling music. In this context, the social dimension of a game offers a natural setting to gauge opinions and collect annotations. The inferior performance we observed for training from expert transcriptions of phone conversations and find a small reduction in performance of the resulting speech recognition system, compared to the same system trained on expert transcriptions. Ambari, et al. (13) also use Mechanical Turk to collect 3,000 English translations of Spanish sentences and train a machine translation system that rivals a system trained on expert annotations.

The proposed game-powered machine learning moves beyond monetary incentives and collects training data for free, a potentially more sustainable and scalable approach. For example, human computation games for annotating multimedia data have succeeded in collecting hundreds of thousands of tags for images (1) and music (4), a significantly larger scale than most Mechanical Turk applications [e.g., thousands to tens of thousands of tags for images (11) or speech transcription (14)]. We investigate whether online annotation games, which entice players with fun and real-time social interaction with a variety of other players, can train systems competitive with those trained from experts. We focus on labeling music. In this context, the social dimension of a game offers a natural setting to gauge opinions and collect annotations. The inferior performance we observed for training from TagATune vs. expert data (see Table S3 at the end of the SI Text) suggested to design and investigate a new game that is adapted to and actively integrated with the machine learning system to improve overall accuracy. Table S3 confirms the resulting system compares well to training from expert data.

**Herd It Minigames**

In order to collect information about diverse aspects of the music as well as to enhance engagement with players, Herd It features a variety of minigames that prompt the Herd to describe the music they hear by:

- catching floating bubbles that describe emotions or instruments present in the song,
- weighing responses to yes/no questions on a scale,
- selecting the most appropriate subgenre from a grid,
- plotting emotional valence and arousal intensity on a Cartesian plane (3, 28) and
- choosing the color that best matches the music.

Each minigame requires a single mouse-click for players to indicate their chosen tag. In addition to the bubbles game depicted in Fig. 2 in the main text, screenshots in Figs. S1 and S2 illustrate the remaining Herd It minigames.

Following each minigame, the player can earn 20 bonus points by correctly naming the song or the artist they have been listening to in a multiple-choice trivia round (see Fig. S3 for an illustration). The sequence of one minigame and one trivia round is repeated for five different songs for a total of 10 rounds. At the end of 10 rounds (lasting 2–3 min), a summary screen presents the final scores, lists the songs that were played during the game and encourages players to connect with the Herd and the rest of their social network.

**Growing The Herd**

To ensure we would generate sufficient Herd It participation to make the game-powered machine learning system viable, we engaged in a user-centered design process (23) to examine the effect of a variety of design features aimed at making the game fun, popular, and possibly viral. Our primary goal in this formative design process was to create a core gameplay experience that was understood by the majority of players and discover problems with the interface that would prevent reliable data collection. We also aimed to build a social gameplay experience that would entertain players and encourage them to share with their friends. Over a 10-month period we conducted regular user-studies both in our lab and in controlled online environments. Each test included a new, previously untested group of between 5 and 50 subjects and focused on free-form issues-based metrics to identify and prioritize crucial issues that were detracting from the user experience (e.g., interviewer observed user mistakes or user verbally expressed frustration or confusion during or after a test session) as well as structured self-reported metrics to provide a quantitative evaluation of the overall experience (e.g., user filled out a questionnaire immediately after a test session) (24). To examine the effect of repeated experiences with Herd It, we also conducted follow-up evaluations with some of the previous test subjects. These follow-up tests determined that users found the game easier to play and more enjoyable during second or subsequent play sessions.

Design issues were identified using a series of in-person interviews and free-form email feedback from online test subjects. Issues-based player quotes included:

- *It took me a while to figure out the agree-a-meter.*
- *The timer was done before I even knew what the game really was. Need clearer instructions—especially the XY minigame.*
- *The interface was kinda busy—it would be nice to have some kind of demo.*
- *I could play the tutorial but not get into the game.*
- *Stuck on “connecting to the herd”.*
- *The song was playing, but the screen was blank.*
- *I could view the demo and instructions but none of the genre tabs opened.*

---

*http://mturk.amazon.com/*
I heard a hiphop song when it was supposed to play Jimi Hendrix’s Star Spangled Banner. The two last songs didn’t play. I clicked well within the time, but it said I did not click before time ran out.

The interviewers’ notes from these sessions included the following observations:

- the timer is a little fast
- the game slowed down with every player
- the scales game needs more questions
- on the XY game, 2 players consistently placed their mark on the axis line
- a player was clicking outside the clickable area on the XY game
- login is frustrating and turning players off

This iterative process identified numerous challenges, inspired design solutions and tested the effect of these redesigns on users. In particular, we detail the development timeline of some of the innovative design features that were found to be crucial for improving interface usability and gameplay efficacy:

Month 1: compute player scores from percentage agreement with the rest of the Herd, illustrated with an “Agree-O-Meter,” rather than an arbitrary scoring metric.

Month 2: replace the generic results screen used for all minigames with customized feedback animations for each minigame to show players how the Herd voted.

Month 3: integrate players’ existing personal data and social network via Facebook, rather than requiring to create a new identity on Herd It.

Month 4: include “name-that-tune” trivia round after each minigame, both to inform players about songs they hear and just for fun.

Month 8: precache all audio clips and game files to minimize gameplay latency.

We aimed to create an enjoyable, positive experience for Herd It players so as to maximize time spent playing and user uptake (i.e., grow the Herd), and thereby the amount of data collected. Social features inspired by this formative design process enabled players to:

Month 5: recommend Herd It, share scores and issue challenges to Facebook friends,

Month 6: chat in real-time with members of the Herd,

Month 9: share music discovered during the game with Facebook friends.

In addition to free-form reporting of issues like those above, at the end of each testing session we asked each test subject to complete an online questionnaire that collected self-reported metrics about the overall experience. More specifically, the subjects responded to the questions listed in Table S1 on a 3- or 5-point Likert scale (25). Subjects also evaluated five of the Herd It minigames after each test, on a discrete scale of five ratings (“Great,” “Good,” “OK,” “Bad,” “Awful”; see Fig. S4). This self-reported data was collected after each of six user tests, spanning 10 min of user-centered design and testing.

Table S1 and Fig. S4 show subject responses from the final user test, at which point we had confidence that players who tried the game would likely understand Herd It and contribute meaningful data. To show how each iteration of the formative design process had a direct impact on the development of Herd It, we also chart the evolution of two key user metrics in Figs. S5 and S6 (over the 10 min of user-centered design). In the second test (conducted on the first day of month 3) we see that our initial reworking of the scoring metric and the improved player feedback developed after the first test had a positive effect on the user experience. Facebook integration was added after that (throughout month 3) and introduced some technical bugs and design challenges that caused some metrics to suffer during the two subsequent user tests. For the fifth test (conducted on the first day of month 8), we experimented with hosting the audio content on a 3rd party server which caused a lot of latency and dropped audio clips, leading to numerous complaints and worse test results. The latency problem inspired the solution (implemented during month 8) of preloading all songs before the game started. Once these challenges were overcome, the final test before launching Herd It (consisting of 50 subjects) showed that 80–90% of the players evaluated these two metrics as “Good” or “Great,” while the number of test subjects giving ratings of “Bad” or “Awful” was reduced to zero. Considering all results in Table S1 and Fig. S4, we see that the majority of users consistently rated their experience as “Good” or “Great” in the final prelaunch test, and said they were likely to share the game with their friends or challenge friends’ high scores.

After almost a year of controlled testing was complete and the core design goals were met, Herd It was launched to the public. Once online, the most current beta version of Herd It underwent continuous, larger-scale testing. The game was exposed to over 200 online users who provided feedback actively (web surveys, emails) and passively via live-site metrics (24) (e.g., ratio of new visitors who registered, clickthrough rates, number of games played, time on site). During this phase of the design process, the core gameplay experience remained unchanged. We focused on enabling and testing features designed to catalyze the continuous recruitment of human music labelers, i.e., (i) acquire new users and (ii) encourage existing users to return and play more games. For example, in order to track progress and save demographic details, a new player arriving at herdit.org was required to “add” the Facebook application before playing Herd It. Although this registration step was very simple (one button click: similar to adding a Facebook friend), it proved to be a barrier as not all players understood Herd It or why they should share their personal information. Our user tests determined that we achieved almost twice as many new players ultimately registered when they were launched directly into a short demonstration game, rather than being required to register immediately. In addition to instructing players on the rules of Herd It, this scripted demo was chosen to include well-known, popular songs and fun tags that would appeal to a wide audience and entice new users to continue playing. Once registered, a new player was brought to the Herd It home page. Multiple iterations of the home page design revealed that simplicity is key: to maximize time spent playing, the home page offers just a few simple buttons that immediately launch the user into the game. Our tests found that new users were 40% more likely to play a second game, once we streamlined the home page. All ancillary features (high score tables and statistics, friend invites, more information about Herd It, music search, etc.) were removed to secondary pages accessed from a list of tabs.

Having enticing a new player to join the Herd, a number of design and gameplay features were included to offer deeper content that would encourage them to return and invite their friends to join. Indeed, one of the motivations for the social elements in Herd It’s design (i.e., multiple simultaneous players, group-based scoring, Facebook integration, sharing of songs and scores, etc.) was to aid in the viral distribution of the game. For example, players were prompted to invite their friends at the end of a game where they accomplished certain achievements (e.g., setting a high-score, advancing in rank or surpassing a friend’s score). Increasing ranks were awarded as users scored more points (e.g., “beginner,” “rock star,” “hip hop hero”) and a scoreboard page tracked each user’s progress daily, weekly and monthly and compared to their friends. Users could post clips of the songs they had enjoyed while playing Herd It on their Facebook wall, sharing the...
musical experience with their friends. Finally, a blog described some of the science behind Herd It and polled users about suggested improvements to the game.

Once the design process was complete and the game was launched, Herd It was promoted to a wide audience of wouldbe players. We leveraged a number of external promotional channels, including: personal emails to friends and coworkers; viral promotion through players’ social networks (suggesting Facebook friends to invite, issuing high-score challenges to friends and sharing songs on Facebook wall); affiliate promotion by inviting musicians to include their songs in Herd It and then promote the game to their fans; media articles and interviews, both in print and online (e.g., blogs, technology news sites).

**Automatic Music Tagging**

Machine learning approaches to modeling the association between semantic tags and spectral patterns in a musical waveform include discriminative learning algorithms (29–35), unsupervised learning algorithms (36), and generative models (31, 37–41). Of these approaches, generative models are generally better suited to handling weakly labeled data (i.e., where songs are labeled only with the presence of some relevant tags) because they estimate audio feature distributions that naturally emerge around audio content relevant to a tag, while down-weighting irrelevant outliers. Furthermore, probabilistic rankings of relevant songs for a given query tag emerge naturally from a generative model.

One of the music autotaggers used in this work, which is also the focus of our active learning approach, is implemented using the generative mixture learning model of ref. 37, based on Gaussian mixture models (GMMs). This model gave rise to a top performing automatic music tagger in the 2008 MIREX evaluation (26). After collecting training data (with some data collection forming automatic music tagger in the 2008 MIREX evaluation (26)), the associations between a vocabulary of tags, \( \mathcal{Y} \), and a training song, \( \mathcal{X} \), are represented as \( y = (y_1, \ldots, y_{|\mathcal{Y}|}) \) where \( y_i > 0 \) if the tag \( w_i \) has been positively associated with the audio of \( \mathcal{X} \) (e.g., if the consensus of Herd It players agrees that the tag \( w_i \) is a good description for the song) and \( y_i = 0 \) otherwise. Fig. S7A shows an example of a group of songs that are all described with the tag “romantic.” Spectral feature vectors \( \chi \) extracted from the audio waveform at regular time intervals, represent a song as a collection of vectors, or “bag of features,” \( \mathcal{X} = \{\chi_1, \ldots, \chi_T\} \), where \( T \) is proportional to the length of the song (Fig. S7B). The system learns a GMM of the audio features for each song, using the standard expectation-maximization (EM) algorithm (42) (Fig. S7C). These song-level GMMs are then combined efficiently into a tag-level GMM using the hierarchical EM algorithm of (43) (Fig. S7D). The result is a model of \( P(\chi|w) \), the distribution of acoustic features \( \chi \) associated with tag \( w_j \):

\[
P(\chi|w_j) = \sum_{n=1}^{N} a_{n}^{w_j} \phi(\chi|\mu_{n}^{w_j}, \Sigma_{n}^{w_j}),
\]

where \( \phi(\cdot|\mu, \Sigma) \) is a multivariate Gaussian distribution with mean \( \mu \) and covariance \( \Sigma \), and the mixture weights \( a_{n}^{w} \) are such that \( a_{n}^{w} \geq 0, \forall n \) and \( \sum_{n} a_{n}^{w} = 1 \). In this work, we use \( N = 16 \) component GMMs to model each tag.

To label a new song \( \mathcal{X} \) using the vocabulary of tags, modeled as above, the likelihood of the bag of features that represents the entire song is inferred under the learned tag-level models using the naïve Bayes assumption of independence between features:

\[
P(\mathcal{X}|w_j) = \prod_{t=1}^{T} P(\chi_t|w_j) \quad \text{(Fig. S8C)}.
\]

Posterior probabilities of each tag, for the new song \( \mathcal{X} \), are found using Bayes’ rule:

\[
P(w_j|\mathcal{X}) = \frac{P(\mathcal{X}|w_j)P(w_j)}{P(\mathcal{X})}.
\]

where \( P(w_j) \) is the prior probability that tag \( w_j \) will appear in an annotation and is assumed to be uniform; \( P(\mathcal{X}) = 1/|\mathcal{Y}| \). The song prior, \( P(\mathcal{X}) \), is obtained by summing the song likelihoods over all \( |\mathcal{Y}| \) tags in the vocabulary:

\[
P(\mathcal{X}) = \sum_{j=1}^{|\mathcal{Y}|} P(\mathcal{X}|w_j)P(w_j).
\]

The final result is a set of semantic weights, \( P(w_j|\mathcal{X}) \), \( \forall w_j \in \mathcal{Y} \), probabilities that suggest how well each tag in the vocabulary describes the song’s acoustic content. The semantic weights for each tag are collected in a semantic multinomial, a probability distribution that provides a rich description of the acoustic content of a song (Fig. S8D). While the alternative autotagging algorithms examined in the main paper [i.e., (32, 33, 41)] use different models of the acoustic content associated with each tag, they each allow to compute a similar probabilistic description of the semantics of a song’s content. Given a semantic query, based on a tag or set of tags, the relevant dimensions of the semantic multinomials are selected to automatically rank songs by their relevance to the query.

**Automatic Tagging Examples**

Table S2 presents results for seven example songs, randomly chosen from personal music collections and never presented to the system before. Each song is analyzed using the GMM autotagger trained on all 127 tags for which Herd It has collected at least 10 reliable examples and then described by inserting the most likely tag or set of tags. The machine learning models trained on Herd It data reliably label new music with a variety of tags. In general, musically objective tags (e.g., “hip hop” and “disco”) are well modeled by machine learning while the subjective tags collected by Herd It’s more whimsical minigames are harder for machine learning models to predict (e.g., the color evoked by the music or songs that are “atmospheric” or “sexy”).

**Music Data**

In this section, we describe in more detail the data used in our experiments, including the audio features, the MGP data and the CAL500 data.

**Audio Features.** The method in ref. 37 used Mel-frequency cepstral coefficients (MFCCs) (44) to capture the spectral content of short-time segments (approximately 5 ms) from each song. For the GMM autotagger used in this work, we instead use the timbre coefficients computed with the feature extraction application programming interface (API) offered online by EchoNest.com, and described at http://developer.echonest.com/docs/method/get_segments/. This open API produces audio descriptors very similar in content to MFCCs but combines feature values over longer-time windows of homogenous audio (variable length but approximately 250 ms), resulting in a more concise representation of each song (i.e., 100’s vs. 10,000’s of feature vectors per song). Tingle, et al. recently showed that these EchoNest timbre feature vectors outperform MFCC feature vectors on the task of automatic music tagging when using the GMM-based system (45). As a result, we use this feature representation and similarly find a (slight) improvement in performance. For the machine learning methods that represent a song as a single feature vector [e.g.,
SVM (33) and Boosting (32), we follow (33) and represent each song as the concatenated mean and variance of its EchoNest timbre features. For the DTM model, which requires a constant interval between extracted feature vectors, we use the MFCCs described above.

Note: To augment content-based music search (based on features that describe the acoustic content only), a wealth of metadata [e.g., artist and song names, web search results (46), lyrics, song reviews, artist biographies, chart position, playlists, etc.] can be collected. To take advantage of this additional information to augment music search, one possible direction that has been explored, for example, is the use of kernel-based methods that can learn from multiple kernels. Such approaches combine kernels derived from the acoustic waveform with kernels derived from metadata extracted online, when available, improving the performance of using either data source in isolation [e.g., (34, 47)]. However, while metadata is readily available for popular songs, in the case of undiscovered songs, where a music search engine would be most useful, this information can not be relied upon (e.g., new songs have not yet been reviewed, unknown artists do not have lyrics or biographies available online). Thus, to satisfy the most general use case (including undiscovered music), our game-powered machine learning solution focuses solely on the only data that is guaranteed to be available: the acoustic waveform.

Training on Expert Labels: The Music Genome Project. The Music Genome Project (MGP), a subsidiary of the Internet radio station Pandora.com, employs musicologists to annotate music using comprehensive surveys. Over the past ten years and at a cost of many millions of dollars, the MGP has labeled hundreds of thousands of songs with up to 500 tags. While these labels are presumably of very high quality, the MGP lexicon is static: adding new styles of music or translating to another language requires laboriously retagging all songs with the new vocabulary and, as musical styles change, the tags can not be easily updated (e.g., contrast tags used to describe “classical” music with those relating to “hip hop” or even “hip hop” in 1981 with “hip hop” in 2011). Although we do not have access to the complete MGP data, we have collected a few MGP tags for 10,000 songs, retrieving publicly available information displayed on Pandora.com (45). Training machine learning models on this MGP subset allows to evaluate our music retrieval system when trained on data derived from a small group of expert labelers.

Evaluating on CAL500. At the University of California San Diego’s Computer Audition Laboratory (CAL), we collected ground-truth data, similar to the Music Genome Project, by paying undergraduate music students to listen to songs and complete a survey that labeled the songs with relevant tags. The CAL500 dataset consists of 500 songs annotated with 149 tags from categories related to musical genres, emotional content, instrumentation, vocal characteristics, and activities during which one might listen to the song (37). Each song was annotated by at least three individuals. It is important to note that, unlike most data mined from online sources, the CAL500 dataset includes both positive and negative associations between songs and tags. Thus CAL500 is suitable for evaluating the output of automatic tagging systems as it is possible to judge correct—and also incorrect—results.

The CAL500 songs were used only for testing the machine learning models; none were included in the training sets. Each learned model is used to rank all the CAL500 songs by their relevance to the tag. For example, the model for the tag “jazz” orders songs by how well they match patterns common to the jazz music in the training set. The ranking is evaluated in reference to the CAL500 ground-truth labels by computing the top-ten precision, the proportion of relevant songs among the first ten results (e.g., the true number of “jazz” songs in the top-ten). High top-ten precision means that many songs at the top of the machine-ranked list are appropriate results for the query tag—exactly the goal of a music search engine.

Dataset Availability. The song labels from all datasets collected for this work—Herd It, MGP and CAL500—are available from the CAL website: http://cosmal.ucsd.edu/cal/projects/AnnRet/.

While copyright issues preclude distribution of the audio files used in this work, all audio features are also available for download.

Comparison To Other Music Annotation Games
The first principal experiment in the main paper, comparing systems based on Herd It vs. MGP data, demonstrated that consensus derived from multiple non-experts playing a human annotation game can train a multimedia retrieval system as reliably as data from expert labelers. With Herd It, we designed a unique, custom solution as opposed to using existing music annotation games such as TagATune (4) or MajorMiner (2) as a source of training data. In particular, TagATune connects pairs of players and asks them to describe the music they hear by typing tags on their computer keyboard. Based on their partner’s tags, players must guess if they are listening to the same song or not. Likewise, MajorMiner’s tagging is entirely user-driven. Herd It, on the other hand, is based on tag suggestion (and confirmation by users), to focus human effort on a larger, more varied vocabulary than TagATune or MajorMiner—which tend to focus on a small set of short, obvious words like, e.g., “drums,” “rock,” “guitar,” etc. Moreover, while the aim of previous games was to have music directly labeled by humans, we designed Herd It explicitly to collect training data for labeling music with machine learning. By suggesting tags for players to confirm, rather than asking players to input tags, Herd It allows to focus data collection on tags (and songs) that are of most interest to the modeling process, in a dynamic way (enabling an active learning loop, as opposed to TagATune’s or MajorMiner’s more open-ended data collection).

To validate the motivation for designing Herd It as a new game, we empirically compare GMM-based models trained on TagATune and Herd It data. TagATune has released a dataset similar to Herd It’s: 10,000 song clips labeled with 188 user-generated tags. MajorMiner data was not explored as it is a private evaluation set for the MIREX challenge (26). Evaluation of GMM-based autotagging models of the 14 tags that are common to TagATune, Herd It, MGP and CAL500 demonstrates that, even with fewer examples of each tag, groups of Herd It players provide more reliable training data than do pairs of TagATune players (see Table S3).

†http://blog.pandora.com/faq
‡Datasets that do not explicitly label negative associations between tags and songs are weakly labeled; the absence of a label may mean that the tag is truly not relevant or that no data was available for this song-tag pair.
§If there are only $n < \frac{1}{2}$ relevant songs in the ground-truth, only the top $n$ results are evaluated.
∥Other metrics that evaluate the complete ranking, such as mean average precision or the area under the receiver operating characteristic curve, exhibit qualitatively similar trends to the results reported here.
¶Other metrics that evaluate the complete ranking, such as mean average precision or the area under the receiver operating characteristic curve, exhibit qualitatively similar trends to the results reported here.
http://tagatune.org/Magnatagatune.html
Fig. S1. Examples of Herd It’s minigames that (A) weigh the Herd’s response to a yes/no question and (B) determine the most appropriate subgenre for a song.

Fig. S2. More examples of Herd It’s minigames that (A) collect two-dimensional valence and arousal on a Cartesian plane and (B) enquire about the color evoked by the music.

Fig. S3. Illustration of Herd It’s trivia round. (A) Players can earn 20 bonus points for correctly naming the song that they hear. (B) The correct song is revealed after the player makes a choice. User testing determined that the familiar “name-that-tune” challenge of the trivia round encouraged novice players to participate in Herd It and the objective, “right/wrong” scoring provided a compelling counterpoint to the subjective, consensus-based scoring of the minigames.
**Fig. S4.** Test subject evaluation of five of the Herd It minigames during the final user test.

**Fig. S5.** Evolution of test subject responses to the question “Overall how much did you like the game?” during 10 m of user-centered design and development.

**Fig. S6.** Evolution of test subject responses to the question “Would you play the game again?” during 10 m of user-centered design and development.

**Fig. S7.** Training an automatic music tagger. (A) Human labelers provide a training set of songs that have been reliably labeled with a given tag (e.g., “romantic”). (B) A “bag of features” represents the waveform of each training song. (C) A GMM of the acoustic features of each song is learned using the EM algorithm (42). (D) The GMMs for each song are combined efficiently into a single GMM that models the acoustic features predictive of the tag using a hierarchical EM algorithm (43).

**Fig. S8.** Automatically tagging a new song. (A) A new, unlabeled song is to be analyzed by the automatic tagging system. (B) A “bag of features” represents the waveform of the song. (C) The features are compared to previously learned models of each tag. (D) Tag probabilities are obtained, providing a semantic profile that describes the song’s acoustic content.
Table S1. Self-reported metrics collected from questionnaires after each of 6 user tests over 10 m

<table>
<thead>
<tr>
<th>Question</th>
<th>Subject response</th>
</tr>
</thead>
<tbody>
<tr>
<td>How much do you play Internet games?</td>
<td>Hard core</td>
</tr>
<tr>
<td></td>
<td>4.0%</td>
</tr>
<tr>
<td>How do you like each minigame? (5 questions)</td>
<td>great</td>
</tr>
<tr>
<td></td>
<td>very</td>
</tr>
<tr>
<td>Are you aware of other people playing with you?</td>
<td>4.0%</td>
</tr>
<tr>
<td>Do you understand how the scores are calculated?</td>
<td>28.0%</td>
</tr>
<tr>
<td>Is music necessary in the game?</td>
<td>100.0%</td>
</tr>
<tr>
<td>Overall, how much did you like the game?</td>
<td>57.2%</td>
</tr>
<tr>
<td>Would you play the game again?</td>
<td>47.0%</td>
</tr>
<tr>
<td>Would you recommend the game to your friends?</td>
<td>43.8%</td>
</tr>
<tr>
<td>Would you try to beat a friend's high-score if you saw it on your Facebook homepage?</td>
<td></td>
</tr>
</tbody>
</table>

Results are shown from the final test with 50 subjects.

Table S2. Automatic music summaries produced by GMM-based machine learning models trained on Herd It data

<table>
<thead>
<tr>
<th>Song</th>
<th>Automatic annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wham! “Careless whisper”</td>
<td>This soft rock song features male lead vocals, feels mellow, evokes the color white and would be good to listen to on a rainy day.</td>
</tr>
<tr>
<td>Neil Young “Heart of gold”</td>
<td>This folk song features bass guitar, feels slow, evokes the color orange and would be good to listen to late at night.</td>
</tr>
<tr>
<td>Michael Jackson “The way you make me feel”</td>
<td>This disco song features drum set, feels catchy, evokes the color yellow and would be good to listen to at dusk.</td>
</tr>
<tr>
<td>Metallica &quot;One&quot;</td>
<td>This rock song features male lead vocals, feels atmospheric, evokes the color orange and would be good to listen to in the morning.</td>
</tr>
<tr>
<td>Lady Gaga “Poker face”</td>
<td>This hip hop song features drum set, feels happy, evokes the color red and would be good to listen to at a party.</td>
</tr>
<tr>
<td>The Flying Burrito Brothers “White line fever”</td>
<td>This folk-rock song features piano, feels acoustic, evokes the color orange and would be good to listen to in the morning.</td>
</tr>
<tr>
<td>Eminem “Kill you”</td>
<td>This hip hop song features drum machine, feels sexy, evokes the color black and would be good to listen to late at night.</td>
</tr>
</tbody>
</table>

For each song, the tags in italics are automatically determined by the automatic tagging system to be the most appropriate genre, instrument, emotion, color and time.

Table S3. Comparison of the average number of training examples available from various data sources and the resulting music tagging performance

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Songs per tag</th>
<th>Top-ten precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagATune(4)</td>
<td>88</td>
<td>0.368</td>
</tr>
<tr>
<td>MGP(45)</td>
<td>851</td>
<td>0.422</td>
</tr>
<tr>
<td>Herd It</td>
<td>46</td>
<td>0.424</td>
</tr>
</tbody>
</table>

Top-ten precision, averaged over the 14 tags in common between all three data sources and CAL500, is evaluated in reference to the CAL500 ground-truth, after training GMM-based models for each tag. While collecting the fewest number of songs for each tag, Herd It clearly provides more reliable examples for training machine learning models than TagATune. Models trained on examples collected by Herd It perform significantly better than those learned from TagATune data (paired t-test, 95% significance level).