Large-Scale Content-Based Audio Retrieval from Text Queries

Gal Chechik
Corresponding Author
Google
gal@google.com

Eugene Ie
Google
eugeneie@google.com

Martin Rehn
Google
rehn@google.com

Samy Bengio
Google
bengio@google.com

Dick Lyon
Google
dicklyon@google.com

ABSTRACT

In content-based audio retrieval, the goal is to find sound recordings (audio documents) based on their acoustic features. This content-based approach differs from retrieval approaches that index media files using metadata such as file names and user tags.

In this paper, we propose a machine learning approach for retrieving sounds that is novel in that it (1) uses free-form text queries rather sound sample based queries, (2) searches by audio content rather than via textual meta data, and (3) can scale to very large number of audio documents and very rich query vocabulary. We handle generic sounds, including a wide variety of sound effects, animal vocalizations and natural scenes. We test a scalable approach based on a passive-aggressive model for image retrieval (PAMIR), and compare it to two state-of-the-art approaches; Gaussian mixture models (GMM) and support vector machines (SVM).

We test our approach on two large real-world datasets: a collection of short sound effects, and a noisier and larger collection of user-contributed user-labeled recordings (25K files, 2000 terms vocabulary). We find that all three methods achieved very good retrieval performance. For instance, a positive document is retrieved in the first position of the ranking more than half the time, and on average there are more than 4 positive documents in the first 10 retrieved, for both datasets. PAMIR completed both training and retrieval of all data in less than 6 hours for both datasets, on a single machine. It was one to three orders of magnitude faster than the competing approaches. This approach should therefore scale to much larger datasets in the future.

Keywords

content-based audio retrieval, ranking, discriminative learning, large margin, large scale

1. INTRODUCTION

Large-scale content-based retrieval of online multimedia documents becomes a central IR problem as an increasing amount of multimedia data, both visual and auditory, becomes freely available. Online audio content is available both isolated (e.g., sound effects recordings), and combined with other data (e.g., movie sound tracks). Earlier works on content-based retrieval of sounds focused on two main thrusts: classification of sounds to (usually a few high-level) categories, or retrieval of sounds by content-based similarity. For instance, people could use short snippets out of a music recording, to locate similar music. This “more-like-this” or “query-by-example” setting is based on defining a measure of similarity between two acoustic segments.

In many cases, however, people may wish to find examples of sounds but do not have a recorded sample at hand. For instance, someone editing her home movie may wish to add car-racing sounds, and someone preparing a presentation about jungle life may wish to find samples of roaring tigers or of tropical rain. In all these cases, a natural way to define the desired sound is by a textual name, label, or description, since no acoustic example is available[1].

Only a few systems have been suggested so far for content-based search with text queries. Slaney [14] proposed the idea of linking semantic queries to clustered acoustic features. Turnbull *et al.* [15], described a system of Gaussian mixture models of music tracks, that achieves good average precision on a dataset with ∼1300 sound effect files.

Retrieval systems face major challenges for handling real-world large-scale datasets. First, high precision is much harder to obtain, since the fraction of positive examples decreases. Furthermore, as query vocabulary grows, more refined discriminations are needed, which are also harder. For instance, telling a lion roar from a tiger roar is harder than telling any roar from a musical piece. The second hurdle is computation time. The amount of sound data available online is huge, including for instance all sound tracks of user contributed videos on YouTube and similar websites. Indexing and retrieving such data requires efficient algorithms and representations. Finally, user-generated content is inherently noisy, in both labels and auditory content. This is partially due to sloppy annotation but more so because different people use different words to describe similar sounds.

[1]Text queries are also natural for retrieval of speech data, but speech recognition is outside the scope of this work.

Retrieval and indexing of user-generated auditory data is therefore a very challenging task.

In this paper, we focus on large-scale retrieval of general sounds such as animal vocalizations or sound effects, and advocate a framework that has three characteristics: (1) Uses text queries rather than sound similarity. (2) Retrieves by acoustic content, rather than by textual metadata. (3) Can scale to handle large noisy vocabularies and many audio documents while maintaining precision. Namely, we aim to build a system that allows a user to enter a (possibly multi-word) text query, and that then ranks the sounds in a large collection such that the most "acoustically relevant" sounds are ranked near the top.

We retrieve and rank sounds not by textual metadata, but by acoustic features of the audio content itself. This approach will allow us in the future to index massively more sound data, since many sounds available online are poorly labeled, or not labeled at all, like the sound tracks of movies. Such a system is different from other information retrieval systems that use auxiliary textual data, such as file names or user-added tags. It requires that we learn a mapping between textual sound description and acoustic features of sound recordings. Similar approaches have been shown to work well for large-scale image retrieval.

The sound-ranking framework that we propose differs from earlier sound classification approaches in multiple aspects: It can handle a large number of possible classes, which are obtained from the data rather than predefined. Since users are typically interested in the top-ranked retrieved results, it focuses on a ranking criterion, aiming to identify the few sound samples that are most relevant to a query. Finally, it handles queries with multiple words, and efficiently uses multi-word information.

In this paper, we propose the use of PAMIR, a scalable machine learning approach trained to directly optimize a ranking criterion over multiple-word queries. We compare this approach to two other machine-learning methods trained on the related multi-class classification task. We evaluate the performance of all three methods on two real-life and large-scale labeled datasets, and discuss their scalability. Our results show that high-precision retrieval of general sounds from thousands of categories can be obtained even with real-life noisy and inconsistent labels that are available today online. Furthermore, these high-precision results can be scale to very large datasets using PAMIR.

2. PREVIOUS WORK

A common approach to content-based audio retrieval is the query-by-example method. In such a system, the user presents an audio document and is proposed a ranked list of audio documents that are the most "similar" to the query, by some measure. A number of studies thus present comparisons of various sound features for that similarity application. For instance, Wan and Lu [19] evaluate features and metrics for this task. Using a dataset of 409 sounds from 16 categories, and a collection of standard features, they achieve about 55% precision, based on the class labels, for the top-10 matches. The approach we present here achieves similar retrieval performance on a much larger dataset, while letting the user express queries using free-form text instead of an audio example.

Turnbull *et al.* [15] described a system that retrieves audio documents based on text queries, using trained Gaussian mixture models as a multi-class classification system. Most of their experiments are specialized for music retrieval, using a small vocabulary of genres, emotions, instrument names, and other predefined "semantic" tags. They recently extended their system to retrieve sound effects from a library, using a 348-word vocabulary [16]. They trained a Gaussian mixture model for each vocabulary word and provided a clever training procedure, as the normal EM procedure would not scale reasonably for their dataset. They demonstrated a mean average precision of 33% on a set of about 1300 sound effects with five to ten label terms per sound file. Compared to their system, we propose here a highly scalable approach that still yields similar mean average precision on much larger datasets and much larger vocabularies.

3. MODELS FOR RANKING SOUNDS

The content-based ranking problem consists of two main subtasks: First, we need to find a compact representation of sounds (features) that allows to accurately discriminate different types of sounds. Second, given these features, we need to learn a matching between textual tags and the acoustic representations. Such a matching function can then be used to rank sounds given a text query.

We focus here on the second problem: learning to match sounds to text tags, using standard features for representing sounds (MFCC, see Sec. 4.2 for a motivation of this choice). We take a supervised learning approach, using corpora of labeled sounds to learn matching functions from data.

We describe below three learning approaches for this problem, chosen to cover the leading generative and discriminative, batch and online approaches. Clearly, the best method should not only provide good retrieval performance, but also scale to accommodate very large datasets of sounds.

The first approach is based on Gaussian mixture models (GMMs) [13], being a common approach in speech and music processing literature. It was used successfully in [16] over a similar content-base audio retrieval task, but using a smaller dataset of sounds and a smaller text vocabulary. The second approach is based on support vector machines (SVMs) [17], being the main discriminant approach in the machine learning literature for classification. The third approach, PAMIR [8], is an online discriminative approach that achieved superior performance and scalability in the related task of content-based image retrieval from text queries.

3.1 The Learning Problem

Consider a text query q and a set of audio documents A, and let $R(q, A)$ be the set of audio documents in A that are relevant to q. Given a query q, an optimal retrieval system should rank all the documents $a \in A$ that are relevant for q ahead of the irrelevant ones;

$$rank(q, a^+) < rank(q, a^-) \quad \forall a^+ \in R(q, A), a^- \in \overline{R}(q, A) \quad (1)$$

where $rank(q, a)$ is the position of document a in the ranked list of documents retrieved for query q. Assume now that we are given a scoring function $F(q, a) \in \mathbb{R}$ that expresses the quality of the match between an audio document a and a query q. This scoring function can be used to order the audio documents by decreasing scores for a given query. Our goal is to learn a function F from training audio documents and queries, that correctly ranks new documents and queries

$$F(q, a^+) > F(q, a^-) \quad \forall a^+ \in R(q, A), a^- \in \overline{R}(q, A) \; . \quad (2)$$

The three approaches considered in this paper (GMMs, SVMs, PAMIR) are designed to learn a scoring function F that fulfills as many of the constraints in Eq. 2 as possible.

Unlike standard classification tasks, queries in our problem often consist of multiple terms. We wish to design a system that can handle queries that were not seen during training, as long as their terms come from the same vocabulary as the training data. For instance, a system trained with the queries "growling lion", "purring cat", should be be able to handle queries like "growling cat". Out-of-dictionary terms are discussed in Sec. 6.

We use the *bag-of-words* representation borrowed from text retrieval [4] to represent textual queries. In this context, all terms available in training queries are used to create a vocabulary that defines the set of allowed terms. This *bag-of-words* representation neglects term ordering and assigns each query a vector $q \in \mathbb{R}^{|T|}$, where $|T|$ denotes the vocabulary size. The t^{th} component q_t of this vector is referred to as the weight of term t in the query q. In our case, we use the *normalized idf* weighting scheme [4],

$$q_t = \frac{b_t^q \; idf_t}{\sqrt{\sum_{j=1}^{|T|} (b_j^q \; idf_j)^2}} \quad \forall t = 1, \ldots, T. \tag{3}$$

Here, b_t^q is a binary weight denoting the presence ($b_t^q = 1$) or absence ($b_t^q = 0$) of term t in q; idf_t is the inverse document frequency of term t defined as $idf_t = -log(r_t)$, where r_t refers to the fraction of corpus documents containing the term t. Here r_t was estimated from the training set labels. This weighting scheme is fairly standard in IR, and assumes that, among the terms present in q, the terms appearing rarely in the reference corpus are more discriminant and should be assigned higher weights.

At query time, a query-level score $F(q, a)$ is computed as a weighted sum of term-level scores

$$F(q, a) = \sum_{t=1}^{|T|} q_t \cdot score_{MODEL}(a, t) \tag{4}$$

where q_t is the weight of the t^{th} term of the vocabulary in query q and $score_{MODEL}()$ is the score provided by one of the three models for a given term and audio document. We now describe separately each of the three models.

3.2 The GMM Approach

Gaussian mixture models (GMMs) have been used extensively in various speech and speaker recognition tasks [12, 13]. In particular, they are the leading approach today for text-independent speaker verification systems. In what follows, we detail how we used GMMs for the task of content-based audio retrieval from text queries.

GMMs are used in this context to model the probability density function of audio documents. The main (obviously wrong) hypothesis of GMMs is that each frame of a given audio document has been generated independently of all other frames; hence, the density of a document is represented by the product of the densities of each audio document frame:

$$p(a|GMM) = \prod_f p(a_f|GMM) \tag{5}$$

where a is an audio document and a_f is a frame of a.

As in speaker verification [13], we first train a single unified GMM on a very large set of audio documents by maximizing the likelihood of all audio documents, using the EM algorithm, without the use of any label. This *background* model can be used to compute $p(a|background)$, the likelihood of observing an audio document a given the background model.

We then train a separate GMM model for each term of the vocabulary T, using only audio documents are relevant for that term. Once trained, each model can be used to compute $p(a|t)$, the likelihood of observing an audio document a given text term t.

Training uses a *maximum a posteriori* (MAP) approach [7], that constrains each term model to stay near the *background* model. The model of $p(a|t)$ is first initialized with the parameters of the background model; Then, the mean parameters of $p(a|t)$ are iteratively modified as in [11]

$$\hat{\mu}_i = \alpha \mu_i^b + (1 - \alpha) \frac{\sum_f p(i|a_f) a_f}{\sum_f p(i|a_f)} \; , \tag{6}$$

where $\hat{\mu}_i$ is the new estimate of the mean of Gaussian i of the mixture, μ_i^b is the corresponding mean in the background model, a_f is a frame of a training set audio document corresponding to the current term, and $p(i|a_f)$ is the probability that a_f was emitted by Gaussian i of the mixture. The regularizer α controls how constrained the new mean is to stay near the background model and is tuned using cross validation.

At query time, the score for a term t and document a is a normalized log likelihood ratio score

$$score_{GMM}(a, t) = \frac{1}{\|a\|} \log \left(\frac{p(a|t)}{p(a|background)} \right) \; , \tag{7}$$

where $\|a\|$ is the number of frames of document a. This can thus be seen as a frame average log likelihood ratio between the term and the background probability models.

3.3 The SVM Approach

Support vector machines (SVMs)[17] are considered to be an excellent baseline system for most classification tasks. SVMs aim to find a discriminant function that maximizes the margin between positive and negative examples, while minimizing the number of misclassifications in training. The trade-off between these two conflicting objectives is controlled by a single hyper-parameter C that is selected using cross validation.

Similarly to the GMM approach, we train a separate SVM model for each term of the vocabulary T. For each term t, we use the training-set audio documents relevant to that term as positive examples, and all the remaining training documents as negatives.

At query time, the score for a term t and document a is as follows:

$$score_{SVM}(a, t) = \frac{SVM_t(a) - \mu_t^{SVM}}{\sigma_t^{SVM}} \quad , \tag{8}$$

where $SVM_t(a)$ is the score of the SVM model for term t applied on audio document a, and μ_t^{SVM} and σ_t^{SVM} are respectively the mean and standard deviation of the scores of the SVM model for term t. This normalization procedure achieved the best performance in a previous study comparing various fusion procedures for multi-word queries [1].

3.4 The PAMIR Approach

The passive-aggressive model for image retrieval (PAMIR) was proposed in [8] for the problem of content-based image retrieval from text queries. It obtained very good performance on this task, with respect to competing probabilistic models and SVMs. Furthermore, it scales much better to very large datasets. We thus adapted PAMIR for retrieval of audio documents and present it below in more detail.

Let query $q \in \mathbb{R}^{|T|}$ be represented by the vector of normalized *idf* weights for each vocabulary term, and a be represented by a vector $\in \mathbb{R}^{d_a}$, where d_a is the number of features used to represent an audio document. Let $\mathbf{W}$ be a matrix of dimensions $(|T| \times d_a)$. We define the query-level score as

$$F_{\mathbf{W}}(q, a) = q^{transp}\mathbf{W}a \ , \tag{9}$$

which measures how well a document a matches a query q. For more intuition, $\mathbf{W}$ can also be viewed as a transformation of a from an acoustic representation to a textual one, $\mathbf{W} : \mathbb{R}^{d_a} \to \mathbb{R}^{|T|}$. With this view, the score becomes a dot product between vector representations of a text query q and a text document $\mathbf{W}a$, as often done in text retrieval [4].

$$score_{PAMIR}(a, t) = \mathbf{W}_t a \tag{10}$$

where $\mathbf{W}_t$ is the t^{th} row of $\mathbf{W}$.

3.4.1 Ranking Loss

Let us assume that we are given a finite training set

$$D_{train} = \left\{(q_1, a_1^+, a_1^-), \ldots, (q_n, a_n^+, a_n^-)\right\}, \tag{11}$$

where for all k, q_k is a text query, $a_k^+ \in R(q_k, A_{train})$ is an audio document relevant to q_k and $a_k^- \in R(q_k, A_{train})$ is an audio document non-relevant to q_k. The PAMIR approach looks for parameters $\mathbf{W}$ such that

$$\forall k, \quad F_{\mathbf{W}}(q_k, a_k^+) - F_{\mathbf{W}}(q_k, a_k^-) \geq \epsilon \ , \quad \epsilon > 0 \tag{12}$$

This equation can be rewritten using the per-sample loss $\forall k$, $l_{\mathbf{W}}(q_k, a_k^+, a_k^-) = \max\left\{0, \epsilon - F_{\mathbf{W}}(q_k, a_k^+) + F_{\mathbf{W}}(q_k, a_k^-)\right\}$. This means that PAMIR looks for $\mathbf{W}$ such that for all k, score $F_{\mathbf{W}}(q_k, a_k^+)$ should be greater than $F_{\mathbf{W}}(q_k, p_k^-)$ by at least a *margin* of ϵ.

This criterion is inspired by the ranking SVM approach [9], which has successfully been applied to text retrieval. However, ranking-SVM requires to solve a quadratic optimization procedure, which does not scale to handle very large number of constraints.

3.4.2 Online Training

PAMIR uses the *passive-aggressive* (PA) family of algorithms, originally developed for classification and regression problems [5] to iteratively minimize

$$L(D_{train}; \mathbf{W}) = \sum_{k=1}^{n} l_{\mathbf{W}}(q_k, a_k^+, a_k^-). \tag{13}$$

At each training iteration i, PAMIR solves the following convex problem:

$$\mathbf{W}^i = \underset{\mathbf{W}}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{W} - \mathbf{W}^{i-1}\|^2 + C \cdot l_{\mathbf{W}}(q_i, a_i^+, a_i^-). \tag{14}$$

where $\|\cdot\|$ is the point-wise L_2 norm. Therefore, at each iteration, $\mathbf{W}^i$ is selected as a trade-off between remaining close to the previous parameters $\mathbf{W}^{i-1}$ and minimizing the loss on the current example $l_{\mathbf{W}}(q_i, a_i^+, a_i^-)$. The *aggressiveness* parameter C controls this trade-off. It can be shown that the solution of problem (14) is,

$$\begin{aligned} & \mathbf{W}^i = \mathbf{W}^{i-1} + \tau_i \mathbf{V}^i, \\ \text{where} \quad & \tau_i = \min\left\{C, \frac{l_{\mathbf{W}^{i-1}}(q_i, a_i^+, a_i^-)}{\|\mathbf{V}^i\|^2}\right\} \\ \text{and} \quad & \mathbf{V}^i = -[q_i^1(a_k^+ - a_k^-), \ldots, q_i^{|T|}(a_k^+ - a_k^-)] \end{aligned}$$

where q_i^j is the j^{th} value of vector q_i, and $\mathbf{V}_i$ is the gradient of the loss with respect to $\mathbf{W}$.

4. EXPERIMENTAL SETUP

We first describe the two datasets that we used for testing our framework. Then we discuss the acoustic features used to represent audio documents. Finally, we describe the experimental protocol.

4.1 Sound Datasets

The success of a sound-ranking system depends on the ability to learn a matching between acoustic features and the corresponding text queries. Its performance strongly depends on the size and type of the sound-recording dataset but even more so on the space of possible queries: classifying sounds into broad acoustic types (speech, music, other) is inherently different from detecting more refined categories such as (lion, cat, wolf).

In this paper we chose to address the hard task of using queries at varying abstraction levels. We collected two sets of data: (1) a "clean" set of sound effects and (2) a larger set of user-contributed sound files.

The first dataset consists of sound effects that are typically short, contain only a single 'auditory object', and usually contain the 'prototypical' sample of an auditory category. For example, samples labeled 'lion' usually contain a wild roar. On the other hand, most sound content that is publicly available, like the sound tracks of home movies and amateur recordings, are far more complicated. They could involve multiple auditory objects, combined into a complex auditory scene. Our second dataset, user-contributed sounds, contains many sounds with precisely these latter properties.

To allow for future comparisons, and since we cannot distribute the actual sounds in our datasets, we have made available a companion website with the full list of sounds for both datasets [2]. It contains links to all sounds available online, and detailed references to CD data, together with the processed labels for each file. This can be found online at *sound1sound.googlepages.com* .

4.1.1 Sound Effects

To produce the first dataset, *SFX*, we collected data from multiple sources: (1) a set of 1400 commercially available sound effects from collections distributed on CDs; (2) a collection of short sounds available from *www.findsounds.com*, including ~3300 files with 565 unique single-word labels; (3) a set of ~1300 freely available sound effects, collected from multiple online websites including *partners in rhyme*, *acoustica.com*, *ilovewavs.com*, *simplythebest.net* , *wav-sounds.com*, *wavsource.com*, *wavlist.com*. Files in these sets usually did not have any detailed metadata except file names.

We manually labeled all of the sound effects by listening to them and typing in a handful of tags for each sound. This was used for adding tags to existing tags (from findsounds) and to tag the non-labeled files from other sources. When labeling, the original file name was displayed, so the labeling decision was influenced by the description given by the original author of the sound effect. We restrict our tags to common terms used in file names, and those existing in the *findsound* data. We also added high level tags to each file. For instance, files with tags such as 'rain', 'thunder' and 'wind' were also given the tags 'ambient' and 'nature'. Files tagged 'cat', 'dog', and 'monkey' were augmented with tags of 'mammal' and 'animal'. These higher level terms assist in retrieval by inducing structure over the label space.

4.1.2 User-contributed Sounds

To produce the second dataset, *Freesound*, we collected samples from the Freesound project [6]. This site allows users to upload sound recordings and today contains the largest collection of publicly available and labeled sound recordings. At the time we collected the data it had more than 40,000 sound files amounting to 150 Gb.

Each file in this collection is labeled by a set of multiple tags, entered by the user. We preprocessed the tags by dropping all tags containing numbers, format terms (mp3, wav, aif, bpm, sound) or starting with a minus symbol, fixing misspellings, and stemming all words using the Porter stemmer for English. Finally, we also filtered out very long sound files (larger than 150 Mb).

For this dataset, we also had access to anonymized log counts of queries, from the freesound.org site, provided by the *Freesound* project. These query counts provide an excellent way to measure retrieval accuracy as would be viewed by the users, since it allows to weight more heavily queries that are popular, and down-weight rare queries.

The query log counts included 7.6M queries. 5.2M (68%) of the queries contained only one term, and 2.2M (28%) had two terms. The most popular query was *wind* (35K instances, 0.4%), followed by *scream* (28420) and *rain* (27594). To match files with queries, we removed all queries that contained non-English characters and the negation sign (less than 0.9% of the data). We also removed format suffixes (*wav*, *aif*, *mp3*), non-letter characters and stemmed query terms similarly to file tags. This resulted in 7.58M queries (622K unique queries, 223K unique terms).

Table 1 summarizes the various statistics for the first split of each dataset, after cleaning.

4.2 Acoustic Features

There has been considerable work in the literature on designing and extracting acoustic features for sound classification (e.g. [19]). Typical feature sets include both time- and frequency-domain features, such as energy envelope and distribution, frequency content, harmonicity, and pitch. The most widely used features for speech and music classification are mel-frequency cepstral coefficients (MFCC). Moreover, in some cases MFCCs were shown to be a sufficient representation, in the sense that adding additional features did not improve classification accuracy [3]. We believe that high-level auditory object recognition and scene analysis could benefit considerably from more complex features and sparse representations, but the study of these features and representations is outside the scope of the current study.

	Freesound	SFX
Number of documents	15780	3431
for training	11217	2308
for test	4563	1123
Number of queries	3550	390
Avg. # of rel. doc. per q.	28.3	27.66
Text vocabulary size	1392	239
Avg. # of words per query	1.654	1.379

Table 1: Summary statistics for the first split of each of the two datasets.

We therefore chose to focus in this work on MFCC-based features. We calculated the standard 13 MFCC coefficients together with their first and second derivatives, and removed the (first) energy component, yielding a vector of 38 features per time frame. We used standard parameters for calculating the MFCCs, as set by the default in the RASTA matlab package, resulting in that each sound file was represented by a series of a few hundreds of 38 dimensional vectors. The GMM based experiments used exactly these MFCC features. For the (linear) SVM and PAMIR based experiments, we wish to represent each file by a single sparse vector. We therefore took the following approach: we used k-means to cluster the set of all MFCC vectors extracted from our training data. Based on small-scale experiments, we settled on 2048 clusters, since smaller numbers of clusters did not have sufficient expressive power, and larger numbers did not further improve performance. Clustering the MFCCs transforms the data into a sparse representation that can be used efficiently during learning.

We then treated the set of MFCC centroids as "acoustic words", and viewed each audio file as a "bag of acoustic words". Specifically, we represented each file using the distribution of MFCC centroids. We then normalized this joint count using a procedure that is similar to the one used for queries. This yields the following acoustic features:

$$a_c = \frac{tf_c^a \ idf_c}{\sqrt{\sum_{j=1}^{d_a} (tf_j^a \ idf_j)^2}}, \tag{15}$$

where d_a is the number of features used to represent an audio document, tf_c^a is the number of occurrences of MFCC cluster t in audio document a, and idf_c is the inverse document frequency of MFCC cluster c, defined as $-log(r_c)$, r_c being the fraction of training audio documents containing at least one occurrence of MFCC cluster c.

4.3 The Experimental Procedure

We used the following procedure for all the methods compared, and for each of our two datasets tested. We used two levels of cross validation, one for selecting hyper parameters, and another for training the models.

Specifically, we first segmented the underlying set of audio documents into three equal non-overlapping splits. Each split was used as a held-out test set for evaluating algorithm performance. Models were trained on the remaining two-thirds of the data, keeping test and training sets always non-overlapping. Reported results are averages over 3 split sets.

To select hyper parameters for each model, we further segmented each training set into 5-fold cross validation sets. For the GMM experiments, the cross-validation sets were used to tune the following hyper-parameters: the number of Gaussians of the background model (tested between 100 and 1000, final value is 500), the minimum value of the variances of each Gaussian (tested between 0 and 0.6, final value is

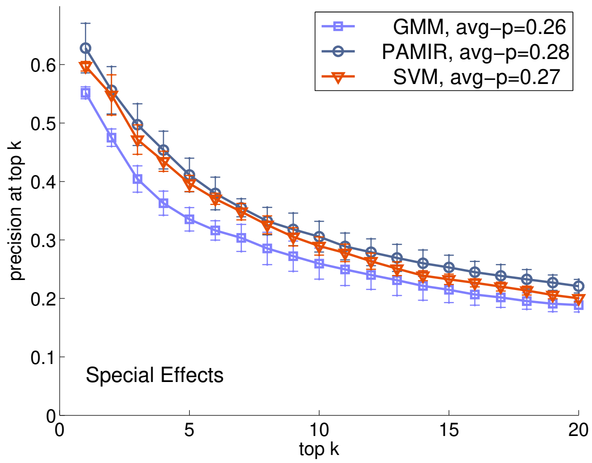

Figure 1: Precision as a function of rank cutoff, SFX data. Error bars denote standard deviation over three split of the data. Avg-p represents the mean average precision for each method.

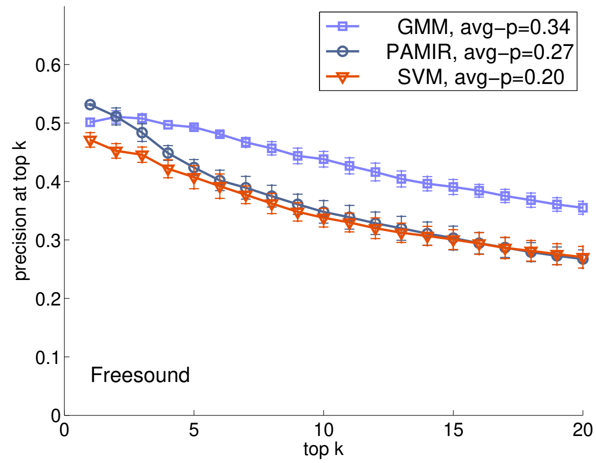

Figure 2: Precision as a function of rank cutoff, Freesound data. Error bars denote standard deviation over three split of the data. All methods achieve similar top-1 precision, but GMM outperforms other methods for precision over lower ranked sounds. On average, eight of the sounds (40%) ranked at top 20 were tagged with labels matched by the query. Avg-p represents the mean average precision for each method.

10^{-9} times the global variance of the data), and α in (6) (tested between 0.1 and 0.9, final value is 0.1). For the SVM experiments, we used a linear kernel and tuned the value of C (tested 0.01, 0.11, 10, 100, 200, 500, 1000, final value is 500). Finally, for the PAMIR experiments, we tuned C (tested 0.01, 0.1, 0.5, 1, 2 10, 100, final value 1).

We used actual anonymized queries submitted to the $Freesound$ database to build the set of queries. An audio file was said to match a query if all its tags are covered by the query. For example, if a document was labeled with tags 'growl' and 'lion', all the queries 'growl', 'lion', 'growl lion' were considered as matching the document. However, a document labeled 'new york' is not matched by a query 'new' or 'york'. All other documents were marked as negative.

The labels in the training set of audio documents were used to define a vocabulary of textual words. We removed from the test sets all queries that were not covered by this vocabulary. We did similar pruning of the validation sets, as constructed from the training sets. Out-of-vocabulary terms are discussed in Sec. 6.

Evaluations

For all experiments, we first compute the per-query precision at top k, defined as the percentage of relevant audio documents within the top k positions of the ranking for the query. Results are then averaged over all queries of the test set, using the observed weight of each query in the query logs. We also report the mean average-precision for each method.

5. RESULTS

We trained the GMM, SVM and PAMIR models on both Freesound and SFX data, and tested their performance and running times. Figure 1 shows the precision at top k as a function of k, for the SFX dataset. All three methods achieve high precision at top-ranked documents with PAMIR outperforming other methods (but not significantly), and GMM providing worse precision. The top-ranked document was relevant to the query in more than 60 % of the queries.

Similarly precise results are obtained for Freesound data, although this data set has an order of magnitude more documents and these are tagged with a vocabulary of query terms that is an order of magnitude larger (Fig. 2). PAMIR is superior for the top 1 and top 2, but then outperformed by GMM, whose precision is consistently higher by ∼10% for all $k > 2$. On average 8 files out of the top 20 are relevant for the query with GMM, and 6 with PAMIR. PAMIR outperforms SVM, although being 10 times faster, and is 400 times faster than GMM on this data.

5.1 Error Analysis

The above results provide average precision across all queries, but queries in our data are highly variable in the number of relevant training files per query. For instance, the number of files per query in Freesound data ranges from 1 to 1049, with most queries having only few files (median = 9). The full distribution is shown in Fig. 3(top). A possible result is that some queries do not have enough files for training on, and hence performance on such poorly sampled queries will be low.

Figure 3(bottom) demonstrates the effect of training sample size per query, within our dataset. It shows that ranking precision greatly improves when more files are available for training, but this effect saturates with ∼ 20 files per query.

We further looked into specific rankings of our system. Since the tags assigned to files are partial, it is often the case that a sound may match a query by its auditory content, but the tags of the file do not match the query words. Table 2 demonstrate this effect showing the 10 top-ranked sound for the query *bigcat*. All first five entries are correctly retrieved, and found precise since their tags contain

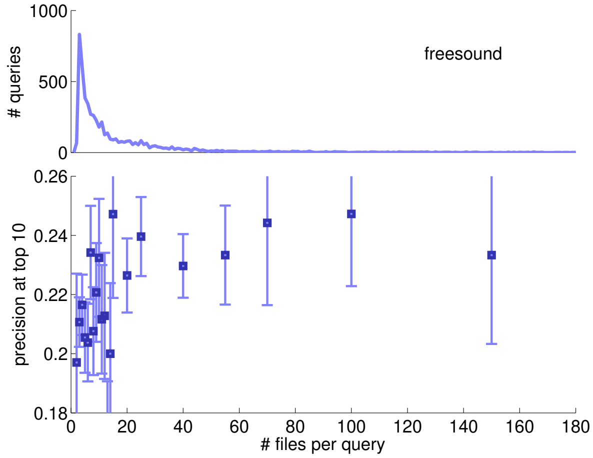

Figure 3: Top: Distribution of number of matching files per query in the training set, Freesound data. Most queries have very few positive examples; mode = 3, median = 9, mean = 28. Bottom: Precision at top 5 as a function of number of files in the training set that match each query, Freesound data. Queries that match less than 20 files, yield on average lower precision. Precision obtained with PAMIR, averaged over all three splits of the data.

Table 2: Top-ranked sounds for the query *big cat*. The file 'guardian' does not match the label 'bigcat' but its acoustic content does match the query.

human eval	eval by tags	score	file name	tags
+	+	4.04	panther -roar2	panther, bigcat, animal, mammal
+	+	3.68	leopard4	leopard, bigcat, animal, mammal
+	+	3.67	panther3	panther, bigcat animal, mammal
+	+	3.53	jaguar	jaguar, bigcat animal, mammal
+	+	3.40	cougar5	cougar, bigcat animal, mammal
+	−	3.26	**guardian**	**growl**, expression, animal
+	+	2.76	tiger	tiger, bigcat animal, mammal
+	+	2.72	Anim-tiger	tiger, bigcat animal, mammal
−	−	2.69	bad disk x	cartoon, synthesized
−	−	2.68	racecar 1	race, motor, engine, ground, machine

Table 3: Total experimental time (training+test) times, in hours, assuming a single modern CPU, for all methods and both datasets, including all feature extraction and hyper-parameter selection. File and vocabulary sizes are for a single split as in Table 1.

Data	files	terms	GMMs	SVMs	PAMIR
Freesound	15780	1392	2400 hrs	59 hrs	6 hrs
SFX	3431	239	960 hrs	5 hrs	3 hrs

the word 'bigcat'. However, entry number 6 is tagged 'growl' (by findsound.com), and was ranked by the system as relevant to the query. Listening to the sound, we confirm that the recording contains the sound of a growling big cat (such as a tiger or a lion). We provide this example online at [2] for the readers of the paper to judge the sounds. This example demonstrates that the actual performance of the system may be considerably better than estimated using precision over noisy tags. Obtaining a quantitative evaluation of this effect requires to carefully listen to thousands of sound files, and is outside the scope of the current work.

5.2 Scalability

The datasets handled in this paper are significantly larger than in previous published work. However, the set of potentially available unlabeled sound is much larger, including for instance sound tracks of user-generated movies available online. The run-time performance of the learning methods is therefore crucial for handling real data in practice.

Table 3 shows the total experimental time necessary to provide all results for each method and database, including feature extraction, hyper-parameter selection, model training, query ranking, and performance measurement. As can be seen, PAMIR scales best, while GMMs are the slowest method for our two datasets. In fact, since SVMs are quadratic with respect to the number of training examples, we expect much longer training times as the number of documents grows to a web scale. Of the methods that we tested, in their present form, only PAMIR would therefore be feasible for a true large-scale application.

For all three methods, adding new sounds for retrieval is computationally inexpensive. Adding new term can be achieved by learning a specific model for the new term, which is also feasible. Significant changes in the set of queries and relevant files may require to retrain all models, but initialization using the older model can make this process faster.

6. DISCUSSION

We developed a scalable system that can retrieve sounds by their acoustic content, opening the prospect to search vast quantities of sound data, using text queries. This was achieved by learning a mapping between textual tags and acoustic features, and can be done for a large open set of textual terms. Our results show that content-based retrieval for general sounds, spanning acoustic categories beyond speech and music, can be accurately achieved, even with thousands of possible terms and noisy real-world labels. Importantly, the system can be rapidly trained on a large set of labeled sound data, and could then be used to retrieve sounds from a much larger (e.g. Internet) repository of unlabeled data.

We compared three learning approaches for modeling the relation between acoustics and textual tags. The most important conclusion is that good performance can be achieved with the highly-scalable method called PAMIR, that was earlier developed for content-based image retrieval. For our dataset, this approach was 10 times faster than multi-class SVM, and 1000 times faster than a generative GMM approach. This suggests that the retrieval system can be further scaled to handle considerably larger datasets.

We used a binary measure to tell if a file is relevant to a query. In some cases, the training data also provides a a continuous relevance score. This could help training by refining the ranking of mildly vs strongly relevant documents. Continuous ranking measures can be easily incorporated to PAMIR since it is based on comparing pairs of documents. This is achieved by adding constraints on the order of two positive documents with one having a higher relevant score than the other one. To handle continuous relevance with SVM, one would have to modify the training objective from a classification task ("is this document related to this term?") to a regression task ("how much is this document related to this term?"). Any regression approaches can be used, including SVM regression, but it is unclear that they would scale and still provide good performance. Finally, it is not clear how the GMM approach could be modified to handle continuous relevance.

The progress of large-scale content-based audio retrieval is largely limited by the availability of high-quality labeled data. Approaches for collecting more labeled sounds could include computer games [18], and using closed-captioned movies. User-contributed data is an invaluable source of labels, but also has important limitations. In particular, users tend to provide annotations with information that does not exist in the recording. This phenomenon becomes most critical in the vision domain, where users avoid "stating the obvious" and describe the context of an image rather than the objects that appear in it. We observe similar effects in the sound domain. In addition, different users may describe the same sound with different terms, and this may cause under-estimates of the system performance.

This problem is related to the issue of 'out-of-dictionary' searches, where search queries use terms that were not observed during training. Standard techniques for addressing this issue make use of additional semantic knowledge about the queries. For instance, queries can be expanded to include additional terms like synonyms or closely related search terms, based on semantic dictionaries or query logs [10]. This aspect is orthogonal to the problem of matching sounds and text queries and was not addressed in this paper.

This paper focused on the feasibility of a large-scale content-based approach to sound retrieval, and all the methods we compared used the standard and widely used MFCC features. The precision and computational efficiency of the PAMIR system can now help to drive progress on sound retrieval, allowing to compare different representations of sounds and queries. In particular, we are currently testing sound representations based on auditory models, which are intended to capture better some perceptual categories in general sounds. Such models could be beneficial in handling the diverse auditory scenes that can be found in the general auditory landscape.

7. ACKNOWLEDGMENTS

We thank D. Grangier for useful discussions and help with earlier versions of this manuscript; Xavier Serra for discussing, information sharing and support with the freesound database.

8. REFERENCES

[1] A. Amir, G. Iyengar, J. Argillander, M. Campbell, A. Haubold, S. Ebadollahi, F. Kang, M. R. Naphade, A. Natsev, J. R. Smith, J. Tesic, and T. Volkmer. IBM research TRECVID-2005 video retrieval system. In *TREC Video Workshop*, 2005.

[2] Anonymous. http://sound1sound.googlepages.com.

[3] J. J. Aucouturier. *Ten Experiments on the Modelling of Polyphonic Timbre*. PhD thesis, Univ. Paris 6, 2006.

[4] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, England, 1999.

[5] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *J. of Machine Learning Research (JMLR)*, 7, 2006.

[6] Freesound. http://freesound.iua.upf.edu.

[7] J. Gauvain and C. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observation of Markov chains. In *IEEE Trans. on Speech Audio Process.*, volume 2, pages 291–298, 1994.

[8] D. Grangier, F. Monay, and S. Bengio. A discriminative approach for the retrieval of images from text queries. In *European Conference on Machine Learning, ECML, Lecture Notes in Computer Science*, volume LNCS 4212. Springer-Verlag, 2006.

[9] T. Joachims. Optimizing search engines using clickthrough data. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[10] R. Jones, B. Rey, O. Madani, and W. Greiner. Generating query substitutions. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 387–396, New York, NY, USA, 2006. ACM.

[11] J. Mariéthoz and S. Bengio. A comparative study of adaptation methods for speaker verification. In *Proc. Int. Conf. on Spoken Lang. Processing, ICSLP*, 2002.

[12] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. Prentice All, first edition, 1993.

[13] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1–3), 2000.

[14] M. Slaney, I. Center, and C. San Jose. Semantic-audio retrieval. In *ICASSP*, volume 4, 2002.

[15] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Towards musical query by semantic description using the cal500 data set. In *SIGIR '07: 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 439–446, New York, NY, USA, 2007. ACM.

[16] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. In *IEEE Transactions on Audio, Speech and Language Processing*, 2008.

[17] V. Vapnik. *The nature of statistical learning theory*. SpringerVerlag, 1995.

[18] L. von Ahn and L. Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326. ACM Press New York, NY, USA, 2004.

[19] P. Wan and L. Lu. Content-based audio retrieval: a comparative study of various features and similarity measures. *Proceedings of SPIE*, 6015:60151H, 2005.