Audio Tag Annotation and Retrieval Using Tag Count Information

Hung-Yi Lo[1,2], Shou-De Lin[2], and Hsin-Min Wang[1]

[1] Institute of Information Science, Academia Sinica, Taipei
[2] Department of Computer Science and Information Engineering, National Taiwan University, Taipei

Abstract. Audio tags correspond to keywords that people use to describe different aspects of a music clip, such as the genre, mood, and instrumentation. With the explosive growth of digital music available on the Web, automatic audio tagging, which can be used to annotate unknown music or retrieve desirable music, is becoming increasingly important. This can be achieved by training a binary classifier for each tag based on the labeled music data. However, since social tags are usually assigned by people with different levels of musical knowledge, they inevitably contain noisy information. To address the noisy label problem, we propose a novel method that exploits the tag count information. By treating the tag counts as costs, we model the audio tagging problem as a cost-sensitive classification problem. The results of audio tag annotation and retrieval experiments show that the proposed approach outperforms our previous method, which won the MIREX 2009 audio tagging competition.

Keywords: Audio tag annotation, audio tag retrieval, folksonomy, tag count, cost-sensitive learning, cost-sensitive evaluation

1 Introduction

With the explosive growth of digital music available on the Web, organizing and retrieving desirable music from online music databases is becoming an increasingly important and challenging task. Until recently, most research on music information retrieval (MIR) focused on classifying musical information with respect to the genre, mood, and instrumentation. Social tags, which have played a key role in the development of "Web 2.0" technologies, have become a major source of musical information for music recommendation systems. Music tags are free text labels associated with different aspects of a music clip, like the artist, genre, emotion, mood, and instrumentation [4]. Consequently, music tag classification seems to be a more complete and practical means of categorizing musical information than conventional music classification. Given a music clip, a tagging algorithm can automatically predict tags for the clip based on models trained from music clips with associated tags collected beforehand.

Automatic audio tag annotation has become an increasingly active research topic in recent years [3, 5, 7, 9, 10], and it has been one of the evaluation tasks at the Music Information Retrieval Evaluation eXchange (MIREX) since 2008[3]. Our previous method [5], which won the MIREX 2009 audio tag annotation competition, exploited both homogeneous segmentation and classifier ensemble techniques. The runner-up [7] in 2009 viewed the audio tag prediction task as a multi-label classification problem and used a stacked (two-stage) SVM to solve it. Another submission [3] introduced a method called the Codeword Bernoulli Average (CBA) model for tag prediction. It is based on a vector quantized feature representation. In the MIREX 2008 evaluation task, the winning audio tag annotation and retrieval system [10] modeled the feature distribution for each tag with a Gaussian mixture model (GMM). The model's parameters were estimated with the weighted mixture hierarchies expectation maximization algorithm. In this paper, our objective is to improve our 2009 winning method by using tag count information.

The audio tagging task can be evaluated from two perspectives: audio tag annotation and audio tag retrieval, as mentioned in [5]. The audio annotation task is viewed as a binary classification problem of each tag, since a fixed number of tags are given. Each tag classifier determines whether the input audio clip should have a specific tag by outputting a score. The performance can be evaluated in terms of the percentage of tags that are verified correctly, or the area under the receiver operating characteristic curve (AUC) given clip (i.e., the correct tags should receive higher scores). In the audio retrieval task, given a specific tag as a query, the objective is to retrieve the audio clips that correspond to the tag. This can be achieved by using the tag classifier to determine whether, based on the score, each audio clip is relevant to the tag. The clips are then ranked according to the relevance scores, and those with the highest scores are returned to the user. The performance can be evaluated in terms of the tag F-measure or the tag AUC.

Social tagging, also called *folksonomy*, enables users to categorize content collaboratively by using tags. Unlike the classification labels annotated by domain experts, the information provided in social tags may contain *noise* or *errors*. Table 1 shows some examples of audio clips with associated tags obtained from the MajorMiner [6] website[4], a web-based game for collecting music tags. Some other details, such as the song's title, album, and artist, are also available. Consider that the tag count indicates the number of users who have annotated the given audio clip with the tag. We believe that tag count information should be considered in automatic audio tagging because the count reflects the confidence degree of the tag. Take the first audio clip from the song *Hi-Fi* as an example. It has been annotated with "drum" nine times, with "electronic" three times and with "beat" twice. Therefore, the tag "drum" is the *major property* of the audio clip. The count also reflects the popularity of the tag, song, artist, and album. To the best of our knowledge, tag count information has not been used in automatic audio tagging previously. In the MIREX audio tagging competition, the tag count is transformed into 1 (with a tag) or 0 (without a tag), by using

[3] http://www.music-ir.org/mirex/2008
[4] http://www.majorminer.org/

a threshold. Then, a binary classifier is trained for each tag to make predictions about untagged audio clips. As a result, a tag assigned twice is treated in the same way as a tag assigned hundreds of times. In addition, a tag with a small count may contain noisy information, which would affect the training of the tag classifier. To solve the problem, we propose using the tag count information to train a cost-sensitive classifier that minimizes the training error associated with tag counts.

Another issue is how to evaluate the prediction performance that considers the tag counts. For example, a system that gives a single tag "drum" to the *Hi-Fi* audio clip should be considered better than a system that gives both "electronic" and "beat" to the clip, but misses "drum". In this paper, we propose two cost-sensitive metrics: a cost-sensitive F-measure and a cost-sensitive AUC. The metrics favor a system that recognizes repeatedly assigned tags.

The contribution of this paper is twofold. First, we employ a cost-sensitive learning approach that treats the tag counts as costs in the audio tag annotation and retrieval task. Second, we propose two cost-sensitive evaluation metrics for the performance evaluation. The results of experiments demonstrate that the proposed cost-sensitive methods outperform their cost-insensitive counterparts in terms of not only the cost-sensitive metrics but also the regular metrics.

The remainder of this paper is organized as follows. In Section 2, we consider some factors that affect the tag counts; and in Section 3, we discuss the proposed cost-sensitive learning methods and cost-sensitive evaluation metrics. In Section 4, we describe the experiments and analyze the results. Section 5 contains some concluding remarks.

Table 1. Some Examples of Audio Clips with Associated Tags Obtained from the MajorMiner Website

Song	Album	Clip Start Time	Artist	Associated Tags (Tag Counts)
Hi-Fi	Head Music	0:00	Suede	drum (9) electronic(3), beat(2)
Universal Traveler	Talkie Walkie	4:00	Air	synth(7), electronic(4), vocal(5) female(4), voice(2), slow(2) ambient(2), soft(3), r&b (3)
Safe	Travis	1:00	The Invisible Band	guitar(5),male(4),pop(4) vocal(3),acoustic(2)
Moritat	Saxophone Colossus	0:50	Sonny Rollins	jazz(9) saxophone(12)
Pacific Heights	Ascension	2:30	Pep Love	male(4), synth(2) hip hop(8), rap(6)
Trouble	The Chillout	3:40	Coldplay	male(6), pop(3), vocal(5) piano(7), voice(3) slow(2), soft(2), r&b(2)

2 Tag Counts

From our study of audio tagging websites, such as Last.fm[5] and MajorMiner, we observe that certain factors affect the tag counts:

1. **Consistent Agreement**: Social tags are usually assigned by users (including malicious users) with different levels of musical knowledge and different intentions [4]. Tags may contain a lot of noisy information; however, when a large number of users consider that an audio clip should be associated with a particular tag, i.e., the count of the tag is high, the label information is deemed more reliable. Conversely, if a tag is only assigned to an audio clip once, the annotation is considered untrustworthy. The MajorMiner website does not show such tags because they may contain noise. When training a classifier, using noisy label information can affect the generalization ability of the classifier. Another problem that must be considered is that, sometimes, only a small portion of an audio clip is related to a certain tag. For example, an instrument might only be played in the bridge section of a song. In this case, the count of the tag that corresponds to the instrument will be small.
2. **Tag Bias**: There are several types of audio tags, e.g., genre, instrumentation, mood, locale, and personal usage. Some types (such as genre) are used more often than others (such as personal usage tags like "favorite" and "I own it"); and specific tags (such as "British rock") are normally used less often than general tags (such as "rock"). In addition, audio tags typically contain many variants [4]. For example, on the Last.fm website "female vocalists" is a common tag, and "female vocals" and "female artists" are variants of it. Figure 1 shows the histogram of the average tag count estimated from MajorMiner data. We observe that the average count of most tags is close to 2.5. The tags with higher average counts are assigned more often than the other tags. The top three most repeatedly assigned tags are "jazz", "saxophone", and "rap"; and the tags assigned least repeatedly are "drum machine", "voice", and "keyboard." We believe that repeatedly assigned tags are either more popular or they describe acoustic characteristics that are easier to recognize (e.g., "drum machine" might easily be recognized as "drum").
3. **Song/Album/Artist Popularity**: Popular songs, albums, and artists usually receive more tags, since people tend to tag music that they like or they are familiar with. However, this is not the case for web-based labeling games because the label flow can be controlled by the game designer. In addition, newly released songs usually receive fewer tags.

Based on the above observations, we formulate the audio tag annotation and retrieval task as a cost-sensitive classification problem. In the next section, we examine the concept of cost-sensitive learning and describe our approach.

[5] http://www.last.fm/

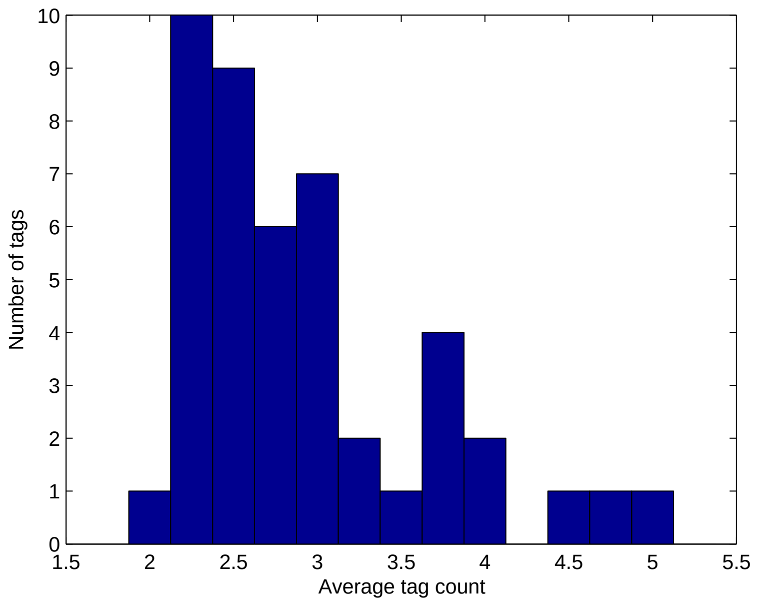

Fig. 1. The histogram of the average tag count

3 Cost-sensitive Learning

Non-uniform misclassification costs are very common in a variety of pattern recognition applications, such as medical diagnosis, e-mail spam filtering, and business marketing. As a result, several cost-sensitive learning methods have been developed to address the problem, e.g., the modified learning algorithm [8] and the data re-sampling method [11]. Suppose we have a cost-sensitive training set $(\boldsymbol{x}_i, y_i, c_i)_{i=1}^{m}$, where $\boldsymbol{x}_i \in R^n$ is the feature vector of the i-th training sample; $y_i \in \{1, -1\}$ is the class label; and $c_i \subset [0, \infty)$ is the misclassification cost. The goal of cost-sensitive classification is to learn a classifier $f(\boldsymbol{x})$, that minimizes the expected cost as follows:

$$E[cI(f(\boldsymbol{x}) \neq y)], \tag{1}$$

where $I(\cdot)$ is an indicator function that yields 1 if its argument is true, and 0 otherwise. The expected cost-insensitive cost is defined as:

$$E[I(f(\boldsymbol{x}) \neq y)], \tag{2}$$

which is a special case of (1) where all samples have an equal misclassification cost c.

The Paralyzed Veterans of America (PVA) dataset used in the KDD Cup 1998 Competition is a well-known dataset for the cost-sensitive learning problem.

It contains information about people who have made at least one prior donation to the PVA, as well as the date and amount of each donation. Participants in the above competition were asked to train a predictive model that would be used to choose the donors that should be sent a request for a new donation. Since mailing a request to an individual donor involves some cost, the measure of success is the total revenue derived from the mailing campaign. The task is formulated as a cost-sensitive classification problem in [11]. Each instance is associated with a misclassification cost, which is calculated as the donated amount minus the cost of mailing the request for positive instances, and as the cost of mailing the request for negative instances. The predictive model is trained to minimize the total misclassification cost on the training data and is expected to maximize the total revenue on the test data.

Based on the above concept, we formulate the audio tag prediction task as a cost-sensitive classification problem. Specifically, we minimize the total counts of misclassified tags by treating the tag counts as costs. In other words, our goal is to correctly predict the most frequently used tags, such as tags of consistent agreement, popular tags, and the tags for popular songs/albums/artists. For example, consider the first audio clip from the song *Hi-Fi* in Table 1 and suppose a tag prediction system A only predicts the tag "drum" correctly, while another tag prediction system B predicts two tags "electronic" and "beat" correctly. We consider that system A outperforms system B because the tag "drum" captures the major property of this audio clip.

3.1 Cost-sensitive Evaluation Metrics

The evaluation metrics used in MIREX 2009, namely, the accuracy, F-measure, tag AUC, and clip AUC, did not consider the costs (i.e., tag counts). Moreover, the class distribution of each binary tag classification problem was imbalanced. For example, in the MajorMiner dataset used at MIREX 2009, out of the forty-five tags, only twelve had more than 10% positive instances. Using accuracy as the evaluation metric biases the system towards the negative class. Since these metrics do not take the costs into account, we propose three cost-sensitive metrics. First, we define the cost-sensitive precision (CSP) and the cost-sensitive recall (CSR) as follows:

$$CSP = \frac{\text{Weighted Sum of TP}}{\text{Weighted Sum of TP+Weighted Sum of FP}}, \tag{3}$$

$$CSR = \frac{\text{Weighted Sum of TP}}{\text{Weighted Sum of TP+Weighted Sum of FN}}, \tag{4}$$

where TP, FP, and FN denote the true positive, the false positive, and the false negative, respectively. The weight of each positive instance is assigned as the count of the associated tag. However, assigning a weight to each negative instance is not as straightforward because people do not use negative tags like "non-rock" and "no drum." Therefore, we assign a uniform cost to negative instances and balance the cost between positive and negative classes, i.e., the total cost of the positive instances is the same as that of the negative instances. As a result, the expected CSP of a random guess baseline method will be 0.5. Then, we can define cost-sensitive metrics based on CSP and CSR.

The *cost-sensitive F-measure* can be calculated as follows:

$$2 \times \frac{\mathrm{CSP} \times \mathrm{CSP}}{\mathrm{CSP} + \mathrm{CSP}}. \tag{5}$$

The receiver operating characteristic curve (ROC) is a graphical plot of the true positive rate (recall) versus the false positive rate as the decision threshold varies. The area under the ROC curve (AUC) is often used to evaluate a binary classifier's performance on a class-imbalanced dataset. We can modify the AUC to obtain a *cost-sensitive AUC* by replacing the recall metric with CSP. Then, we use the cost-sensitive clip AUC and the cost-sensitive tag AUC to evaluate the audio annotation task and the audio retrieval task, respectively.

3.2 Cost-sensitive Classification Methods

Support vector machine (SVM) and AdaBoost are two very effective learning algorithms for classification problems. In this subsection, we describe their cost-sensitive versions.

The training process of SVM attempts to maximize the margin and minimize the training error at the same time. The objective function for cost-sensitive SVM is formulated as follows:

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},b,\xi} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum\limits_{i=1}^{m} c_i\xi_i, \\ \text{s.t.} & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0, i = 1, \ldots, m, \end{array} \tag{6}$$

where the parameter $\boldsymbol{w}$ can be learned by solving a minimization problem; ϕ is a function that maps the input data to a higher dimensional space and C is a tuning parameter that exists in the general SVM form. Note that each cost c_i is associated with a corresponding training error term ξ_i.

AdaBoost finds a highly accurate classifier by combining several base classifiers, even though each of them is only moderately accurate. Cost-sensitive AdaBoost [8] maintains a weight vector D_t for the training instances in each iteration and uses a base learner to find a base classifier to minimize the weighted error according to D_t. In each iteration, the weight vector D_t is updated by

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t c_i y_i h_t(\boldsymbol{x}_i))}{Z_t}, \tag{7}$$

where $h_t(\boldsymbol{x}_i)$ is the prediction score of the base classifier h_t for instance $\boldsymbol{x}_i$; c_i is the cost of each instance; Z_t is a normalization factor that makes D_{t+1} a distribution; and α_t can be calculated based on different versions of AdaBoost. We use a decision stump as the base learner in this study. The other implementation details are the same as the standard AdaBoost [2].

4 Experiments

The experiments follow our previous setup of the MIREX 2009 extended experiments reported in [5]. We consider forty-five tags, which are associated with 2,472 audio clips downloaded from the website of the MajorMiner game. The duration of each clip is 10 seconds or less.

Given an audio clip, we divide it into several homogeneous segments by using an audio novelty curve [1]. Then, using MIRToolbox 1.1[6], we extract a 174-dimensional audio feature vector from each segment to reflect various types of musical information, such as the segment's dynamics, rhythm, timbre, pitch, and tonality. For an audio clip, the prediction score given by a classifier is the average of its constituent segments. The SVM and AdaBoost scores are merged by using either a probability ensemble to annotate an audio clip, or a ranking ensemble to rank all the audio clips according to a tag. For the ranking ensemble, we first rank the prediction scores of individual classifiers independently. Then, a clip's final score is the average of the rankings derived by the two classifiers. For the probability ensemble, we transform the output score of each component classifier into a probability score with a sigmoid function, and then compute the average of the two probability scores.

4.1 Model Selection and Evaluation

We adopt three-fold cross-validation in the experiments. The audio clips are randomly split into three subsets. In each fold, one subset is selected as the test set and the remaining two subsets serve as the training set. The test set for (outer) cross-validation is not used to determine the classifier's settings. Instead, we perform inner cross-validation on the held out data from the training set to determine the cost parameter C in SVM and the number of base learners in AdaBoost. Then, we retrain the classifiers with the complete training set and the selected parameters, and perform outer cross-validation on the test set. We use the AUC as the model selection criterion.

To calculate the tag F-measure, we need a threshold to binarize the output score. In the audio retrieval task, we want to retrieve audio clips from an audio database. We assume that each tag's class has similar probability distributions in the training and testing audio databases; therefore, we set the threshold with the class's distribution obtained from the training data. In the audio annotation task, we annotate the test audio clips one by one. We set the threshold to 0.5 because the calibrated probability score ranges from 0 to 1.

4.2 Experiment Results

We compare the proposed method, which uses the tag count information, with our MIREX 2009 winning method, which did not use such information. The experiment results in terms of the cost-sensitive metrics and regular metrics are

[6] http://users.jyu.fi/ lartillo/mirtoolbox/

summarized in Tables 2 and 3, respectively. The AdaBoost, SVM, and classifier ensemble are the same as those used in the above winning method. We perform three-fold cross-validation twenty times and calculate the mean and standard deviation of the results to reduce the variance caused by different cross-validation splits. Note that the tag AUC and F-measure are more suitable for the audio retrieval task, while the clip AUC is more suitable for the audio annotation task.

Table 2. Evaluation results of different classifiers and ensemble methods in terms of cost-sensitive metrics.

Mean ±St.d.	CS Tag AUC	CS Clip AUC	CS F-measure
AdaBoost	0.8055 ±0.0027	0.8892 ±0.0011	0.4099 ±0.0052
CS AdaBoost	0.8169 ±0.0023	0.8967 ±0.0005	0.4469 ±0.0081
SVM	0.8112 ±0.0022	0.8957 ±0.0007	0.4354 ±0.0077
CS SVM	0.8215 ±0.0023	0.9005 ±0.0004	0.4593 ±0.0056
Ensemble	0.8334 ±0.0019	0.8979 ±0.0007	0.4606 ±0.0067
CS Ensemble	0.8356 ±0.0018	0.9032 ±0.0006	0.4808 ±0.0072

The results in Table 2 demonstrate the effectiveness of using the tag counts to train a cost-sensitive classifier. The cost-sensitive methods outperform their cost-insensitive counterparts. The improvement in the cost-sensitive F-measure is the most significant: 3.7% for CS AdaBoost versus AdaBoost and 2.4% for CS SVM versus SVM. In addition, SVM slightly outperforms AdaBoost, and the ensemble methods outperform the classifiers using SVM or AdaBoost alone.

Table 3 compares the results of different methods in terms of the regular (cost-insensitive) evaluation metrics. Interestingly, the cost-sensitive methods outperform their cost-insensitive counterparts in terms of the regular metrics. Recall that tags with smaller counts may contain *noisy labeling information.* By viewing the tag counts as costs, the cost-sensitive learning method can ignore the noisy information by giving a smaller penalty (cost), and thereby train a more accurate classifier.

5 Conclusion

We have proposed a novel method for exploiting tag count information in audio tagging tasks, and discussed several factors that affect the counts of tags assigned to an audio clip. Among the factors, we believe that consistent agreement

Table 3. Evaluation results of different classifiers and ensemble methods in terms regular (cost-insensitive) metrics.

Mean ±St.d.	Tag AUC	Clip AUC	F-measure
AdaBoost	0.7941 ±0.0027	0.8773 ±0.0011	0.3018 ±0.0035
CS AdaBoost	0.8050 ±0.0023	0.8854 ±0.0005	0.3216 ±0.0049
SVM	0.7992 ±0.0021	0.8837 ±0.0007	0.3226 ±0.0053
CS SVM	0.8106 ±0.0021	0.8894 ±0.0004	0.3299 ±0.0037
Ensemble	0.8221 ±0.0018	0.8859 ±0.0007	0.3386 ±0.0046
CS Ensemble	0.8247 ±0.0017	0.8921 ±0.0005	0.3442 ±0.0046

is the most important issue, since noisy labeling information in tags assigned less frequently impacts the training of a tag classifier. We formulate the audio tag prediction task as a cost-sensitive classification problem in order to minimize the misclassified tag counts. In addition, we present cost-sensitive versions of several regular evaluation metrics. The proposed cost-sensitive methods outperform their cost-insensitive counterparts in terms of not only the cost-sensitive evaluation metrics but also the regular evaluation metrics.

We have realized that the audio tagging task can also be formulated as a multi-label classification problem. In our future work, we will develop a cost-sensitive multi-label learning algorithm for the task. We will also evaluate our methods on more multimedia tagging tasks.

References

1. Foote, J., Cooper, M.: Media segmentation using self-similarity decomposition. In: SPIE (2003)
2. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences 55(1) (1997)
3. Hoffman, M., Blei, D., Cook, P.: Easy as cba: A simple probabilistic model for tagging music. In: ISMIR (2009)
4. Lamere, P.: Social tagging and music information retrieval. Journal of New Music Research 37(2), 101–114 (2008)
5. Lo, H.Y., Wang, J.C., Wang, H.M.: Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval. In: ICME (2010)
6. Mandel, M.I., Ellis, D.P.W.: A web-based game for collecting music metadata. In: ISMIR (2007)

7. S. Ness, A. Theocharis, L.G.M., Tzanetakis, G.: Improving automatic msic tag annotation using stacked generalization of probabilistic svm outputs. In: ACM MM (2009)
8. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40(12), 3358–3378 (2007)
9. Tingle, D., Kim, Y., Turnbull, D.: Exploring automatic music annotation with "acoustically-objective" tags. In: MIR (2010)
10. Turnbull, D., Barrington, L., Torres, D., Lanckriet, G.: Semantic annotation and retrieval of music and sound effects. IEEE Trans. on Audio, Speech, and Language Processing 16, 467–476 (2008)
11. Zadrozny, B., Langford, J., Abe, N.: Cost-sensitive learning by cost-proportionate example weighting. In: ICDM (2003)