Cost-Sensitive Multi-Label Learning for Audio Tag Annotation and Retrieval

Hung-Yi Lo, Ju-Chiang Wang, Hsin-Min Wang, *Senior Member, IEEE*, and Shou-De Lin

***Abstract*—Audio tags correspond to keywords that people use to describe different aspects of a music clip. 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. Our method that won the MIREX 2009 audio tagging competition is one of this kind of methods. However, since social tags are usually assigned by people with different levels of musical knowledge, they inevitably contain noisy information. By treating the tag counts as costs, we can model the audio tagging problem as a cost-sensitive classification problem. In addition, tag correlation information is useful for automatic audio tagging since some tags often co-occur. By considering the co-occurrences of tags, we can model the audio tagging problem as a multi-label classification problem. To exploit the tag count and correlation information jointly, we formulate the audio tagging task as a novel cost-sensitive multi-label (CSML) learning problem and propose two solutions to solve it. The experimental results demonstrate that the new approach outperforms our MIREX 2009 winning method.**

***Index Terms*—Audio tag annotation, audio tag retrieval, tag count, cost-sensitive learning, multi-label.**

I. 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 artist, 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, including locale, opinion, personal usage, etc., in addition to artist, genre, mood, and instrumentation [1]. 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 tagging has become an increasingly active research topic in recent years [2]–[7], and it has been one of the evaluation tasks at the Music Information Retrieval Evaluation eXchange (MIREX) since 2008[1]. We participated in the MIREX 2009 audio tag classification task and our system was ranked first in terms of the tag F-measure and the area under the receiver operating characteristic curve (AUC) given tag [4]. The advantage of our winning method is twofold. First, we divide the audio clip into several homogeneous segments by using an audio novelty curve [8]. Second, we exploit an ensemble classifier, which consists of Support Vector Machine (SVM) and AdaBoost classifiers, for tag classification. This paper starts with a detailed description of our winning method, and then presents several novel techniques to improve it, namely, transforming the output scores of the component classifiers into calibrated probability scores such that they can be easily combined by the classifier ensemble, using the tag count information to train a cost-sensitive classifier that minimizes the training error associated with tag counts, and using multi-label classification to handle tag correlation information. Part of this work appears in a conference paper [4].

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 I shows some examples of audio clips with associated tags obtained from the MajorMiner [9] website[2], 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 [10]. 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

This work was supported in part by the Taiwan e-Learning and Digital Archives Program (TELDAP) sponsored by the National Science Council of Taiwan under Grant: NSC99-2631-H-001-020.

H.-Y. Lo is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan, and also with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan. E-mail: hungyi@iis.sinica.edu.tw.

J.-C. Wang is with the Department of Electrical Engineering, National Taiwan University, Taipei 106, Taiwan, and also with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan. E-mail: asriver@iis.sinica.edu.tw.

H.-M. Wang is with the Institute of Information Science, Academia Sinica, Taipei 115, Taiwan. E-mail: whm@iis.sinica.edu.tw.

S.-D. Lin is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan. E-mail: sdlin@csie.ntu.edu.tw.

[1] http://www.music-ir.org/mirex/2008

[2] http://www.majorminer.org/

TABLE I
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)

tag "drum" captures the *more salient property* of the audio clip than the tags "electronic" and "beat". In addition, a tag with a small count may even 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.

Tag correlation is another useful information for automatic audio tagging since some tags often co-occur. For example, a song with the "hip hop" tag is more likely to be also annotated with "rap" than "jazz", while a song with the "dance" tag is more likely to be also annotated with "electronic" than "guitar". However, previous research [2], [11], [12] usually assumes that the tags are independent and, thus, transforms the tag prediction problem into many independent binary classification problems, each for an individual tag. This manner inevitably loses the co-occurrence information of multiple tags that might be useful for automatic audio tagging. We believe that multi-label classification, in which an instance can be associated with multiple labels, is more suitable for the task than binary classification. Existing multi-label classification approaches can be grouped into two categories: *algorithm adaptation* and *problem transformation* [13]. The first category extends some specific learning algorithms for binary classification to solve the multi-label classification problem, while the second category transforms the multi-label classification problem to one or many single-label classification tasks. To exploit the tag count and correlation information jointly, we formulate the audio tagging task as a novel cost-sensitive multi-label (CSML) learning problem and propose two solutions to solve it. To the best of our knowledge, cost-sensitive multi-label classification has not been studied previously.

A. Related Works

The winning audio tagging system [7] at MIREX 2008 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. The runner-up [5] at MIREX 2009 viewed the audio tag prediction task as a multi-label classification problem and used a stacked SVM to solve it. Another submission [3] in 2009 introduced a method called the Codeword Bernoulli Average (CBA) model for tag prediction. It is based on a vector quantized feature representation.

Using audio segmentation for music genre and artist classification has been studied in [11]. An audio clip was simply partitioned into several *fixed length* segments. In our work, we divide an audio clip into several homogeneous segments by using an audio novelty curve, and then extract audio features from each segment. The features in frame-based feature vector sequence format are further represented by their mean and standard deviation such that they can be combined with other segment-based features.

Current cost-sensitive learning research has been focused on binary or multi-class classification [14], but never on multi-label classification. Although cost-sensitive classification considering non-uniform misclassification costs has been applied in many real-world applications, such as medical diagnosis and fraud detection, *tag count* information has not been well considered 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 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 [9], the authors compared binary classifiers trained on the verified tags (i.e., the tags assigned by at least two people) with that trained on all tags, and found that the former case achieved better performance than the latter. They have also tried different thresholds to select the verified tags, but the classification accuracy did not change much.

Some previous research [5], [15] tried to model the co-occurrences of tags using two-stage methods. In the first stage, the tags are assumed independent, and a binary classifier is trained for each tag. In the second stage, SVM [5] or a Dirichlet mixture model [15] is used to combine the tag classifiers' prediction scores. In [16], the authors used a canonical correlation analysis based method, which projected a label space into a compact subspace that maximized the correlation between the feature space and the label space. Our work is based on two multi-label classification algorithms: stacking [17] and random k-Labelsets (RAkEL) [18]. We extend these two methods for cost-sensitive multi-label classification.

B. Performance Evaluation

The audio tagging task can be evaluated from two perspectives: audio tag annotation and audio tag retrieval. The audio tag 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 AUC given clip (i.e., the correct tags should receive higher scores). In the audio tag 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 per-tag F-measure or per-tag AUC.

An important 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.

C. Contribution

The contribution of this paper is fourfold.

1) We propose dividing the audio clip into several homogeneous segments by using an audio novelty curve, and exploit an ensemble classifier, which consists of SVM and AdaBoost classifiers, for tag classification. This system was ranked first in the MIREX 2009 audio tag classification task, in terms of tag AUC (the average of per-tag AUCs over all tags) and tag F-measure (the average of per-tag F-measures over all tags).
2) We propose transforming the output scores of the component classifiers into calibrated probability scores such that they can be easily combined by the classifier ensemble. This step can improve the performance in terms of clip AUC.
3) We formulate the audio tag annotation and retrieval task as a cost-sensitive multi-label classification problem by treating the tag counts as misclassification costs and considering the co-occurrences of tags, and propose two methods, namely cost-sensitive stacking and cost-sensitive RA*k*EL, to solve it.
4) We propose two cost-sensitive evaluation metrics for the performance evaluation.

The remainder of this paper is organized as follows. In Section II, we give an overview of our audio tag annotation and retrieval system. Then, we describe feature extraction and audio segmentation in Section III, and present our MIREX 2009 winning method in Section IV. In Section V, we consider several factors that affect the tag counts and introduce the concept of cost-sensitive learning. We present the proposed cost-sensitive multi-label classification methods and the cost-sensitive evaluation metrics in Sections VI and VII, respectively. Then, we discuss the results of the MIREX 2009 audio tagging competition and extended experiments in Sections VIII and IX, respectively. Finally, Section X contains some concluding remarks.

II. System Overview

Fig. 1 shows the work flow of our audio tag annotation and retrieval system. We first split an audio clip into homogeneous segments, and then extract audio features with respect to various musical information, including dynamics, rhythm, timbre, pitch, and tonality, from each segment. The features in frame-based feature vector sequence format are further represented by their mean and standard deviation such that they can be combined with other segment-based features to form a fixed-dimensional feature vector for a segment. The prediction score for an audio clip given by a classifier is the average of the scores for its constituent segments. In the training phase, we train classifiers using our proposed cost-sensitive multi-label learning methods. In the testing phase, the classifiers output scores for audio tag annotation and retrieval, respectively.

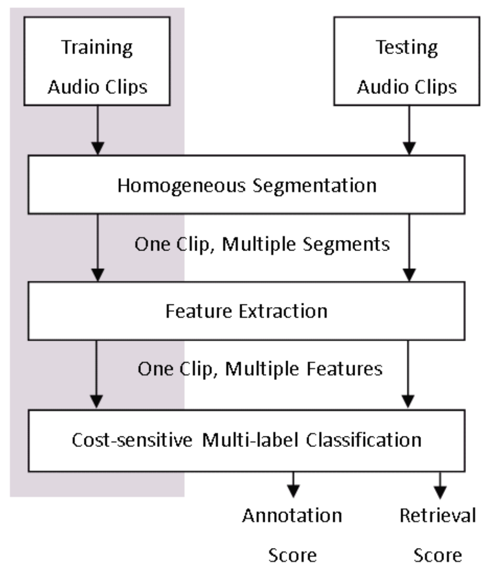

Fig. 1. Work flow of the proposed audio tag annotation and retrieval system.

III. Audio Signal Processing

For applying machine learning techniques to audio tag classification, we need to extract characteristic features of various types from the waveform of an audio clip by using some signal processing methods. Since feature selection is embedded in the training process of our classification method, we extract as many kinds of features as possible. However, for some frame-based features, such as Mel-frequency cepstral coefficients (MFCCs), we need to convert the variable-length feature vector sequence into a fixed-dimensional feature vector such that they can be used jointly with other features, like key and tempo. In this paper, the frame-based features are represented by their mean and standard deviation calculated over the audio clip (or segment as will be discussed later).

It is very likely that only a portion of the audio clip is associated with a specific tag. For instance, an audio clip may have the tag "female vocal" even though a female vocal only appears in the front part of the clip. Therefore, it might be inadequate to use the mean of MFCC vectors to represent the timbre of the whole clip. To solve this problem, we divide the

TABLE II
MUSIC FEATURES USED IN THIS WORK

Category	Feature	Category	Feature
Dynamics	rms	Fluctuation	fluctuation peak
Rhythm	tempo		fluctuation centroid
	attack time	Timbre	zero crossing rate
	attack slope		spectral flux
Spectrum	centroid		low energy rate
	spread		MFCC
	skewness		delta MFCC
	kurtosis		delta-delta MFCC
	entropy	Tonality	key clarity
	flatness		key mode possibility
	rolloff at 85%		harmonic change
	rolloff at 95%		chromagram
	brightness		chroma centroid
	roughness		chroma peak
	irregularity	Pitch	pitch value

clip into homogeneous segments and treat each segment as a unit in tag classification. Then, the final decision for the clip is based on the fusion of the results of its constituent segments.

A. Feature Extraction

To extract music features, we use MIRToolbox 1.1[3], a free software that comprises approximately 50 audio/music feature extractors and statistical descriptors [19]. As shown in Table II, we consider seven categories of features in this work, namely, dynamics, rhythm, spectrum, fluctuation, timbre, tonality, and pitch. We set default values for parameters in MIRToolbox, such as the length of window and hop size. After feature extraction, each clip (or segment) is represented by a 174-dimensional feature vector.

B. Audio Segmentation

Our audio segmentation is based on a measure of audio novelty proposed in [8]. We use the function implemented in MIRToolbox 1.1. An example segmentation result is shown in the bottom panel of Fig. 2. We first compute the cosine measure of MFCC vectors between any pairs of two frames in the audio clip, and build a self-similarity matrix, which can be visualized as a square image in the top panel of Fig. 2. The color scale of a pixel in the image is proportional to the similarity. Then, we can obtain a time-aligned novelty curve, as shown in the middle panel of Fig. 2, by convolving a checkerboard kernel with a radial Gaussian taper along the diagonal of the similarity matrix. The radial Gaussian taper of width H is defined as:

$$t(a,b) = \exp\{-4 \times [(\frac{a-H/2}{H/2})^2 + (\frac{b-H/2}{H/2})^2]\}, \qquad (1)$$

where $a, b = 1, 2, \cdots, H$, are the horizontal and vertical indexes of the Gaussian taper, respectively. Therefore, we only need to calculate a diagonal strip of width H when constructing the similarity matrix. In this paper, H is set to 64. Finally, the local peaks of the novelty curve, as marked by circles in the middle panel of Fig. 2, are selected as segment boundaries. To prevent feature extraction and classification failures caused by insufficient data, we require the length of each segment to be at least 0.5 seconds.

[3]http://users.jyu.fi/~lartillo/mirtoolbox/

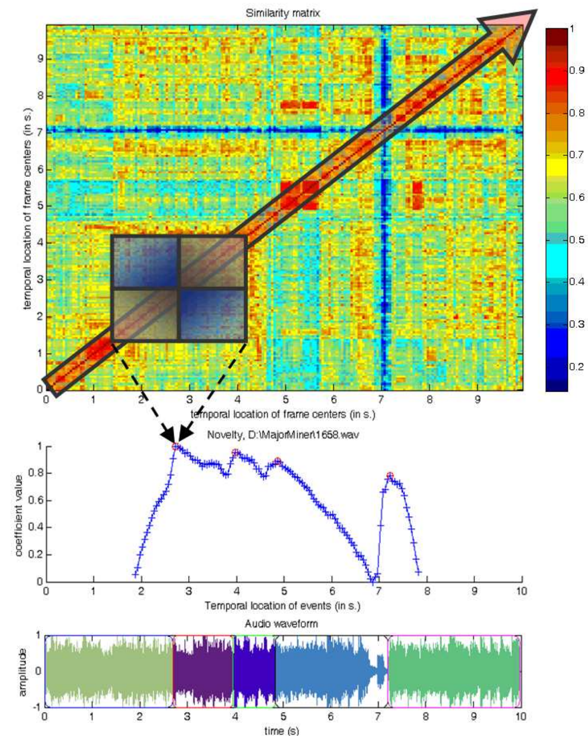

Fig. 2. Illustration of audio segmentation.

IV. THE MIREX 2009 WINNING METHOD

In this section, we discuss our MIREX 2009 winning method. In the method, we assume the tags occur independently and do not consider the co-occurrences of tags. As a result, the tag classification problem is viewed as many independent binary classification problems, with a binary classifier for each individual tag. For each tag, the final prediction combines the outputs of two classifiers: SVM and AdaBoost. The tag count is transformed into 1 (with a tag) or 0 (without a tag) by using a threshold. Fig. 3 shows an overview of the classifier ensemble. It is used instead of the cost-sensitive multi-label classification in Fig. 1.

A. Support Vector Machine

SVM finds a separating surface with a large margin between training samples of two classes in a high-dimensional feature space implicitly introduced by a computationally efficient kernel mapping [20]. The large margin implies good generalization ability according to statistical learning theory. In this work, we exploit a linear SVM classifier $f(\boldsymbol{x})$ of the following form:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b. \qquad (2)$$

Given a training set $(\boldsymbol{x}_i, y_i)_{i=1}^N$ for a specific tag, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the feature vector of the i-th training sample and $y_i \in \{1, -1\}$ is the class label, the parameters

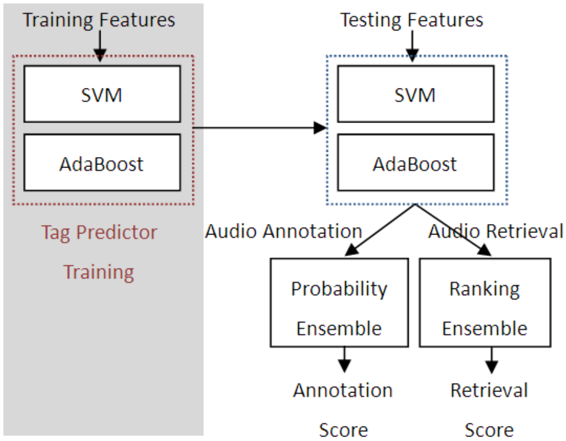

Fig. 3. Work flow of the classifier ensemble used in the MIREX 2009 winning method.

$\boldsymbol{w} = (w_1, w_2, ..., w_d)$ and b can be learned by solving a minimization problem formulated as follows:

$$\begin{array}{lrcl} \min\limits_{\boldsymbol{w},b,\xi} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} & + & C\sum\limits_{i=1}^{N}\xi_i, \\ \text{s.t.} & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) & \geq & 1 - \xi_i \\ & \xi_i & \geq & 0 \end{array} \tag{3}$$

where ξ_i is the training error associated with instance $\boldsymbol{x}_i$; C is a tuning parameter that controls the tradeoff between maximizing the margin and minimizing the training error. The selection of C will be discussed in Section VIII.

B. AdaBoost

AdaBoost [21] finds a highly accurate classifier by combining several base classifiers, even though each of them is only moderately accurate. It has been successfully used in applications such as music classification [11] and audio tag classification [2]. The decision function of the AdaBoost classifier takes the following form:

$$f(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}), \tag{4}$$

where $h_t(\boldsymbol{x})$ is the prediction score of a base classifier h_t given the feature vector $\boldsymbol{x}$ of a test sample; T is the number of base classifiers; and α_t can be calculated based on different versions of AdaBoost.

The base classifiers are learned iteratively. In the training phase, AdaBoost [22] maintains a weight vector D_t for the training instances in each iteration and uses a base learner to find a base classifier h_t to minimize the weighted error according to D_t. In each iteration, the weight vector D_t is updated by

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(\boldsymbol{x}_i))}{Z_t}, \tag{5}$$

where Z_t is a normalization factor that makes D_{t+1} a distribution. We can increase the number of base learners iteratively and stop the training process when the generalization ability on the validation set does not improve. We use a decision stump as the base learner in this study.

C. Ranking Ensemble

We noticed that the scales of the two classifiers' prediction scores are rather different. Given a batch of testing clips, we first rank the prediction scores of individual classifiers independently. Then, the final score for a clip is the average of the ranks from the two classifiers. In this way, the smaller the average rank, the more likely the audio clip has the specific tag. We have applied this method in our MIREX 2009 submission. It achieves very good performance in terms of tag F-measure and tag AUC as these two metrics are more related to the ranking performance. However, the performance in terms of clip AUC is poor. In fact, this method is not suitable for the audio tag annotation task because it is unpractical to annotate a clip by referring to other clips simultaneously. In order to annotate a single clip, we need to combine the scores from the two classifiers in a different way. Therefore, we propose probability ensemble instead of ranking ensemble for the audio tag annotation task.

D. Probability Ensemble

As each tag classifier is trained independently, the raw scores of different tag classifiers are not comparable. In the audio tag annotation task, we need to compare the scores of all tag classifiers and determine which tags should be associated with the given audio clip. Therefore, we transform the output score of SVM or AdaBoost into a probability score with a sigmoid function [23]:

$$Pr(y = 1|\boldsymbol{x}) \approx \frac{1}{1 + \exp(Af(\boldsymbol{x}) + B)}, \tag{6}$$

where $f(\boldsymbol{x})$ is the output score of a classifier on instance $\boldsymbol{x}$, A and B are learned by solving a regularized maximum likelihood problem as suggested in [24]. As the classifier output has been calibrated into a probability score, a classifier ensemble is formed by averaging the probability scores of SVM and AdaBoost, and the probability scores of different tag classifiers become comparable.

V. Tag Counts And Cost-Sensitive Learning

In this section, we describe our idea to improve the above MIREX 2009 winning method by treating the tag counts as costs and applying cost-sensitive learning. We first discuss several factors that affect the tag counts, and then introduce the concept of cost-sensitive learning.

A. Tag Counts

From our study of audio tagging websites, such as Last.fm 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 [1]. 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 as discussed in Section I-A. 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 could 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 [1]. For example, on the Last.fm website "female vocalists" is a common tag, and "female vocals" and "female artists" are variants of it. Fig. 4 shows the histogram of the average tag count estimated from MajorMiner data. As mentioned earlier, the count of each tag is at least 2. We observe that the average counts of most tags are close to 2.5. Some tags have higher average counts 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 describe acoustic characteristics that are easier to recognize (e.g., "drum machine" might easily be recognized as "drum").

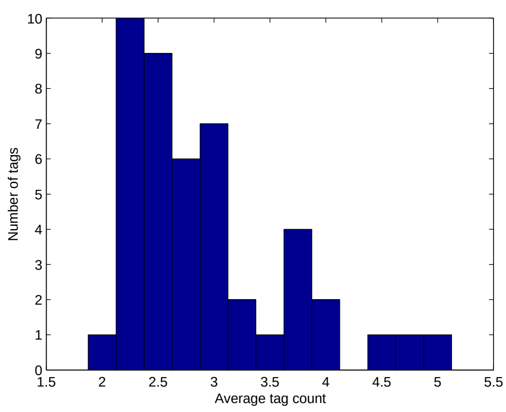

Fig. 4. Histogram of the average counts of tags on the MajorMiner website.

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 some web-based labeling games, such as MajorMiner, 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 subsection, we examine the concept of cost-sensitive learning.

B. *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 [22] and the data re-sampling method [25]. Suppose we are given a cost-sensitive training set $(\boldsymbol{x}_i, y_i, c_i)_{i=1}^{N}$ for a binary classification task, where $c_i \subset [0, \infty)$ 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:

$$E[cI(f(\boldsymbol{x}) \neq y)], \tag{7}$$

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:

$$E[I(f(\boldsymbol{x}) \neq y)], \tag{8}$$

which is a special case of (7) where all samples have an equal misclassification cost c.

To formulate the audio tag prediction task as a cost-sensitive classification problem, 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 I 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 count of "drum" is more than the sum of the other two tags of this audio clip.

Both SVM and AdaBoost can be extended to the cost-sensitive versions to solve the cost-sensitive binary classification problem. Cost-sensitive SVM can be learned by modifying (3) to

$$\begin{array}{cccl} \min\limits_{\boldsymbol{w},b,\xi} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} & + & C\sum\limits_{i=1}^{N} c_i\xi_i, \\ \text{s.t.} & y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) & \geq & 1 - \xi_i \\ & \xi_i & \geq & 0 \end{array} \tag{9}$$

where each cost c_i is associated with a corresponding training error term ξ_i. Cost-sensitive AdaBoost [22] can be learned by modifying the update rule of weight vector D_t in (5) to

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t c_i y_i h_t(\boldsymbol{x}_i))}{Z_t}, \tag{10}$$

where c_i is the cost of training instance $\boldsymbol{x}_i$.

However, cost-sensitive binary classification still assumes the tags are independent and loses the co-occurrence information of tags. Therefore, we propose using cost-sensitive multi-label classification for the audio tagging task in the next section.

VI. Cost-Sensitive Multi-Label Classification

We first introduce the concept of multi-label classification. Let $\boldsymbol{x} \in \mathbb{R}^d$, which is a d-dimensional input space, and $\mathcal{Y} \subseteq \mathcal{L} = \{1, 2, ..., K\}$, which is a finite set of K possible labels. To facilitate the discussion, hereafter, $\mathcal{Y}$ is represented by a vector $\boldsymbol{y} = (y_1, y_2, ..., y_K) \in \{1, -1\}^K$. Given a training set $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$ that contains N samples, the goal of multi-label classification is to learn a classifier $h : \mathbb{R}^d \rightarrow 2^{\mathcal{L}}$ such that $h(\boldsymbol{x})$ predicts which labels should be assigned to an unseen sample $\boldsymbol{x}$.

Cost-sensitive multi-label classification extends multi-label classification by coupling a cost vector $\boldsymbol{c}_i \in \mathbb{R}^K$ to each training sample $(\boldsymbol{x}_i, \boldsymbol{y}_i)$. The j-th component c_{ij} denotes the cost to be paid when the label y_{ij} is misclassified. More specifically, c_{ij} is a *false negative cost* when $y_{ij} = 1$, and a *false positive cost* when $y_{ij} = -1$. In this work, the false negative cost is set as the tag count while the false positive cost is uniformly set to one. We extend two existing multi-label learning algorithms, namely, stacking [17] and RA*k*EL [18], to solve the CSML problem.

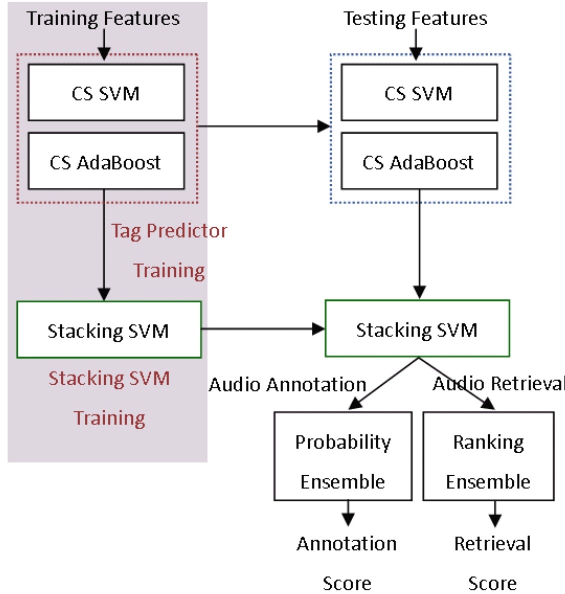

Fig. 5. Work flow of the cost-sensitive stacking-based audio tag annotation and retrieval system.

A. Cost-Sensitive Stacking

Stacking is a method of combining the outputs of multiple independent classifiers for multi-label classification. Assume that the K tags are independent and their tag classifiers are trained independently. The first step of using stacking for multi-label classification is to use the outputs of all classifiers, $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_K(\boldsymbol{x})$, as features to form a new feature set. Let the new feature be $\boldsymbol{z} = (z_1, z_2, ..., z_K)$. Then, we can use the new feature set together with the true label to learn the parameters w_{kj} of the stacking classifiers:

$$h_k(\boldsymbol{z}) = \sum_{j=1}^{K} w_{kj} z_j, \tag{11}$$

where the weight w_{kj} will be positive if tag j is positively correlated to tag k; otherwise, w_{kj} will be negative. The stacking classifiers can recover misclassified tags by using the correlation information captured in the weight w_{kj}.

Inspired by the idea of stacking, we improve our MIREX 2009 classifier ensemble by using cost-sensitive stacking. As shown in Fig. 5, we first train K SVM-based and K AdaBoost-based cost-sensitive binary tag classifiers by using the tag counts as costs independently. Then, we use stacking SVM to respectively process the outputs of the SVM-based binary tag classifiers and the outputs of the AdaBoost-based binary tag classifiers. Note that the stacking SVM itself is cost-insensitive. Finally, we apply the ranking ensemble or probability ensemble method.

B. Cost-Sensitive RA*k*EL

Label powerset (LP) method is a major category of multi-label learning algorithms. It reduces the multi-label classification problem to a single-label multi-class classification problem by treating each distinct combination of labels in the training set as a different class. It is computationally more efficient than treating the multi-label classification problem as several binary classification problems. However, when the number of labels increases, the number of classes increases exponentially, and each class will be associated with very few training instances.

In [18], a method called Random k-Labelsets is proposed to realize the LP method. A k-labelset is a labelset $\mathcal{R} \subseteq \mathcal{L}$ with $|\mathcal{R}| = k$. RA*k*EL randomly selects a number of k-labelsets from $\mathcal{L}$ and uses the LP method to train the corresponding multi-label classifiers. Algorithms 1 and 2 describe the training and classification processes of RA*k*EL, respectively. The prediction of a multi-class LP classifier g_m for sample $\boldsymbol{x}$ is denoted by $g_m(\boldsymbol{x}) \in \{1, 2, \ldots, V\}$. Note that V will be much smaller than 2^k if the data is sparse. In Algorithm 2, $q(g_m(\boldsymbol{x}), j)$ is defined as:

$$q(g_m(\boldsymbol{x}), j) = \begin{cases} 1 & j \in \mathcal{R}_m \text{ and } j \text{ is positive in } g_m(\boldsymbol{x}), \\ -1 & j \in \mathcal{R}_m \text{ and } j \text{ is negative in } g_m(\boldsymbol{x}), \\ \varnothing & j \notin \mathcal{R}_m. \end{cases} \tag{12}$$

For example, when $k = 2$, the classes 1, 2, 3, and 4 correspond to $(1, 1), (1, -1), (-1, 1),$ and $(-1, -1)$, respectively. If tag j is not included in $\mathcal{R}_m$, $q(\cdot, j)$ is undefined. If tag j corresponds to the first label of $\mathcal{R}_m$, $q(1, j), q(2, j), q(3, j),$ and $q(4, j)$ will output scores $1, 1, -1,$ and -1, respectively.

In order to use the tag counts as misclassification costs, we extend RA*k*EL for cost-sensitive multi-label classification. The extension is not straightforward since we are given a cost value for each label but RA*k*EL considers a set of labels as a class. Our idea is to train the cost-sensitive LP classifier $\hat{g}_m$ by transforming the cost of each label in a labelset to the total cost of the labelset. The transformed cost $\hat{c}_i$ of a training

Algorithm 1 The training process of RA*k*EL

- **Input:** number of models M, size of labelset k, set of labels $\mathcal{L}$, and the training set $\mathcal{D} = (\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^N$
- **Output:** an ensemble of LP classifiers g_m and the corresponding k-labelsets $\mathcal{R}_m$

1) Initialize $\mathcal{S} \leftarrow \mathcal{L}^k$
2) **for** $m \leftarrow 1$ **to** min(M,$|\mathcal{L}^k|$) **do**
   - $\mathcal{R}_m \leftarrow$ a k-labelset randomly selected from $\mathcal{S}$
   - train the LP classifier g_m based on $\mathcal{D}$ and $\mathcal{R}_m$
   - $\mathcal{S} \leftarrow \mathcal{S} \setminus \mathcal{R}_m$
3) **end**

Algorithm 2 The classification process of RA*k*EL

- **Input:** number of models M, a test sample $\boldsymbol{x}$, an ensemble of LP classifiers g_m, and the corresponding k-labelsets $\mathcal{R}_m$
- **Output:** the multi-label classification vector $\boldsymbol{r} = (r_1, r_2, ..., r_K)$

1) **for** $j \leftarrow 1$ **to** K **do**
   a) $r_j = 0, n = 0$
   b) **for each** g_m, **if** $j \in \mathcal{R}_m$ **do**
      - $r_j = r_j + q(g_m(\boldsymbol{x}), j)$
      - $n = n + 1$
   c) **end**
   d) $r_j = \frac{r_j}{n}$
2) **end**

sample $\boldsymbol{x}_i$ for training $\hat{g}_m$ is computed by

$$\hat{c}_i(\boldsymbol{c}_i, \boldsymbol{y}_i) = \begin{cases} \sum\limits_{j \in \mathcal{R}_m \text{ s.t. } y_{ij}=1} c_{ij} & \text{if } \exists j \in \mathcal{R}_m \text{ s.t. } y_{ij} = 1, \\ 1 & \text{else,} \end{cases} \tag{13}$$

where $\boldsymbol{c}_i$ is the cost vector mentioned in the beginning of Section VI. Therefore, we can obtain the multi-class training sample with the associated cost, $(\boldsymbol{x}_i, \hat{y}_i, \hat{c}_i)$, for training the LP classifier, where $\hat{y}_i \in \{1, 2, \dots, V\}$ is the class value and $\hat{c}_i$ is the cost to be paid when the class of this instance is misclassified. We use the multi-class SVM as the LP classifier in this study, and employ the one-versus-one strategy [26] in cost-sensitive multi-class classification. In the experiments, we will compare RA*k*EL with cost-sensitive RA*k*EL.

VII. Cost-Sensitive Evaluation Metrics

The evaluation metrics used at MIREX 2009, namely, the accuracy, tag 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:

$$CSP = \frac{\text{Weighted Sum of TP}}{\text{Weighted Sum of TP+Weighted Sum of FP}}, \tag{14}$$

$$CSR = \frac{\text{Weighted Sum of TP}}{\text{Weighted Sum of TP+Weighted Sum of FN}}, \tag{15}$$

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:

$$\frac{2 \times \text{CSR} \times \text{CSP}}{\text{CSR} + \text{CSP}}. \tag{16}$$

The receiver operating characteristic curve is a graphical plot of the true positive rate (recall) versus the false positive rate as the decision threshold varies. The 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 CSR. Then, we use the cost-sensitive clip AUC and the cost-sensitive tag AUC to evaluate the audio tag annotation task and the audio tag retrieval task, respectively.

VIII. MIREX 2009 Results

The submissions to the MIREX 2009 audio tag classification task have been evaluated on two datasets: the MajorMiner set and the mood set [27]. The algorithms were evaluated with three-fold cross validation and artist filtering was used in the production of the test and training splits. The evaluation metrics include the tag F-measure and tag AUC. Both metrics correspond to the tag retrieval task that is aimed at retrieving audio by a given tag query. The metrics also include clip AUC and the tag accuracy. These two metrics correspond to the tag annotation task that is aimed at annotating a given audio clip with correct tags.

The results of evaluation on the two datasets are summarized in Tables III and IV, respectively. The best result of each specific evaluation metric is bold-typed. The names in the first column indicate the twelve submissions. Our submissions without and with pre-segmentation are denoted by NOS and SEG, respectively. It is clear that pre-segmentation is effective. Table V summarizes the ranking of our two submissions in terms of the four evaluation metrics on the two datasets. Our SEG submission achieves the best performance in terms of the metrics corresponding to the audio tag retrieval task (i.e., tag F-measure and tag AUC) but performs poorly in terms of the metric corresponding to the audio tag annotation task (i.e.,

TABLE III
EVALUATION RESULTS OF MIREX 2009 AUDIO TAG CLASSIFICATION ON THE MAJORMINER DATASET. THERE ARE 12 SUBMISSIONS. OUR SUBMISSIONS WITHOUT AND WITH PRE-SEGMENTATION ARE DENOTED BY NOS AND SEG, RESPECTIVELY

	Tag F-measure	Tag Accuracy	Tag AUC	Clip AUC
NOS	28.90	90.01	78.22	75.14
SEG	**31.08**	90.35	**80.73**	77.37
A1	27.68	86.78	74.18	87.08
A2	28.99	85.95	76.14	86.13
B1	20.93	91.22	76.16	**88.24**
B2	24.14	90.51	79.07	88.23
B3	17.05	91.32	72.11	85.45
B4	26.26	88.95	74.85	85.45
C	1.22	89.08		
D1	29.00	85.02	78.39	87.24
D2	29.34	85.05	78.62	87.63
E	4.43	**91.44**	73.64	85.11

TABLE IV
EVALUATION RESULTS OF MIREX 2009 AUDIO TAG CLASSIFICATION ON THE MOOD DATASET. THERE ARE 12 SUBMISSIONS. OUR SUBMISSIONS WITHOUT AND WITH PRE-SEGMENTATION ARE DENOTED BY NOS AND SEG, RESPECTIVELY

	Tag F-measure	Tag Accuracy	Tag AUC	Clip AUC
NOS	20.37	88.21	66.68	67.83
SEG	**21.95**	88.65	**70.12**	70.40
A1	19.49	83.70	64.84	85.39
A2	19.26	82.90	63.19	85.93
B1	17.23	87.77	65.18	84.85
B2	17.98	88.15	68.12	84.85
B3	14.71	88.22	62.88	81.18
B4	18.33	86.17	64.60	81.19
C	8.40	86.27		
D1	21.14	82.32	64.90	86.04
D2	20.88	82.38	65.49	**86.13**
E	6.32	**90.92**	66.45	86.12

clip AUC). The details about the evaluation datasets and the other submissions are available on the MIREX website[4].

IX. EXTENDED EXPERIMENTS

This section presents the results of extended experiments on the downloaded MajorMiner dataset. We extensively evaluate the individual classifiers, the ranking ensemble method, the probability ensemble method, and the CSML methods.

[4]http://www.music-ir.org/mirex/wiki/2009:MIREX2009_Results

TABLE V
PERFORMANCE RANKINGS OF OUR TWO SUBMISSIONS TO MIREX 2009 AUDIO TAG CLASSIFICATION ON TWO DATASETS

	Evaluation Metrics	Ranking SEG	Ranking NOS
The MajorMiner Dataset	Tag AUC	1	5
	Tag F-measure	1	5
	Clip AUC	11	12
	Tag Accuracy	5	6
The Mood Dataset	Tag AUC	1	3
	Tag F-measure	1	4
	Clip AUC	11	12
	Tag Accuracy	2	3

TABLE VI
THE 45 TAGS USED IN THE MIREX 2009 AUDIO TAG CLASSIFICATION EVALUATION

metal	instrumental	horns	piano	guitar
ambient	saxophone	house	loud	bass
fast	keyboard	electronic	noise	british
solo	electronica	beat	80s	dance
jazz	drum machine	strings	pop	r&b
female	rock	voice	rap	male
slow	vocal	quiet	techno	drum
funk	acoustic	distortion	organ	soft
country	hip hop	synth	trumpet	punk

A. Dataset

Our extended experiments basically follow the MIREX 2009 setup. The evaluation data come from the MajorMiner's music labeling game[5], which invites players to listen to short music clips (each about 10 seconds long) and label them with relevant words and phrases. According to the MIREX 2009 audio tag classification result web page, 45 tags, as listed in Table VI, are considered. We download all the audio clips that are associated with these 45 tags from the website of the MajorMiner's game. The resulting audio database contains 2,472 clips and the duration of each clip is 10 seconds or less. The dataset might be slightly different from that used in MIREX 2009 because the MajorMiner website might have been updated recently.

B. Model Selection and Evaluation

We adopt three-fold cross-validation in the experiments following the evaluation method at MIREX 2009. The 2,472 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 for determining the classifier setting. Instead, we first 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 re-train the classifiers with the complete training set and the selected parameters, and perform outer cross-validation on the test set. Since the class distributions for some tags are imbalanced (more than two thousand negative instances and less than fifty positive instances), classification accuracy is not a fair criterion for model selection. Therefore, we use the tag AUC as the model selection criterion. For RA*k*EL and its cost-sensitive version, the number of models, M, is set to 250, and the size of the labelset, k, is set to 14.

To calculate the tag F-measure and tag accuracy, we need a threshold to binarize the output score. For the audio tag retrieval task, we want to retrieve audio clips from the audio database. We assume that each tag's class has similar probability distributions in the training and testing audio databases.

[5]http://majorminer.org/

Therefore, we set the threshold to select relevant audio clips according to the class prior distribution obtained from the training data. For the audio tag annotation task, we annotate the testing audio clips one by one. We set the threshold to 0.5 because the calibrated probability score ranges from 0 to 1.

C. Experimental Results

Our experimental results in terms of the metrics corresponding to the audio tag annotation task and the audio tag retrieval task are summarized in Table VII. Because the cross-validation split used in MIREX 2009 is not available, we perform three-fold cross-validation twenty times and calculate the mean and standard deviation of the results of different cross-validation splits.

Several observations can be drawn from Table VII. First, pre-segmentation is effective. All the classification methods benefit from pre-segmentation. For example, the tag AUC is improved by 1.42% (cf. SVM) and 4.23% (cf. AdaBoost). Second, SVM slightly outperforms AdaBoost. Third, the two ensemble methods are respectively suitable for either the retrieval task or the annotation task as discussed above. On the audio tag retrieval task, ranking ensemble not only has better mean performance than any individual classifier, but also has a smaller standard deviation. Probability ensemble is more suitable than ranking ensemble for the audio tag annotation task. However, the improvement over the SVM classifier is small.

Next, we compare the CSML methods, which exploit the tag count and correlation information jointly, with the MIREX 2009 winning method. We also evaluate the cost-sensitive binary classification (CS only) methods and the cost-insensitive multi-label (ML only) classification methods. We perform three-fold cross-validation one hundred times and calculate the mean and standard deviation of the results. The experimental results in terms of the cost-sensitive metrics and regular metrics are summarized in Tables VIII and IX, respectively. The AdaBoost-, SVM-, and Ensemble-MIREX methods are the same as that used in Table VII. The Ensemble methods use probability ensemble to generate Clip AUC and ranking ensemble to generate F-measure and Tag AUC. For the AdaBoost, SVM, and Ensemble methods, the ML only method employs stacking, and the CSML method employs cost-sensitive stacking. The cost-sensitive RA*k*EL method is compared to its cost-insensitive version.

The results in Table VIII demonstrate the effectiveness of CSML learning. The improvement in the cost-sensitive F-measure is the most significant: 3.87% for AdaBoost-CSML versus AdaBoost-MIREX, 3.66% for SVM-CSML versus SVM-MIREX, 2.27% for Ensemble-CSML versus Ensemble-MIREX, and 1.07% for cost-sensitive RA*k*EL (i.e., RA*k*EL-CSML) versus RA*k*EL (i.e., RA*k*EL-ML only). The cost-sensitive stacking methods outperform their cost-insensitive binary classification counterparts in terms of all evaluation metrics (cf. AdaBoost-CS only versus AdaBoost-MIREX and AdaBoost-CSML versus AdaBoost-ML only). However, cost-sensitive RA*k*EL is slightly worse than RA*k*EL in terms of cost-sensitive Clip AUC and Tag AUC, although the difference is not significant. From the table, we observe that both the CS only methods and the ML only methods are effective and the CS only methods are slightly better than the ML only methods. We also observe that the standard deviations of the results are very small.

TABLE VIII
AUDIO TAG ANNOTATION AND RETRIEVAL RESULTS OF COST-SENSITIVE MULTI-LABEL CLASSIFICATION METHODS IN TERMS OF COST-SENSITIVE METRICS (IN %)

Classifier	Mean ±St.d.	CS Clip AUC	CS F-measure	CS Tag AUC
AdaBoost	MIREX	88.92±0.09	41.04±0.65	80.53±0.24
	CS Only	89.67±0.07	44.71±0.58	81.74±0.21
	ML Only	89.60±0.11	42.71±0.68	80.97±0.32
	CSML	**89.88±0.09**	**44.91±0.62**	**81.79±0.29**
SVM	MIREX	89.47±0.09	43.95±0.56	81.22±0.28
	CS Only	90.08±0.06	45.92±0.56	82.20±0.20
	ML Only	90.08±0.08	45.49±0.51	82.53±0.20
	CSML	**90.67±0.07**	**47.61±0.63**	**83.11±0.23**
Ensemble	MIREX	89.65±0.07	46.04±0.57	83.03±0.19
	CS Only	90.32±0.06	47.94±0.61	83.63±0.18
	ML Only	90.21±0.07	46.65±0.56	83.45±0.18
	CSML	**90.61±0.06**	**48.31±0.62**	**83.89±0.17**
RA*k*EL	ML Only	**91.11±0.08**	45.57±0.59	**84.77±0.12**
	CSML	90.97±0.09	**46.64±0.61**	84.13±0.16

TABLE IX
AUDIO TAG ANNOTATION AND RETRIEVAL RESULTS OF COST-SENSITIVE MULTI-LABEL CLASSIFICATION METHODS IN TERMS OF REGULAR (COST-INSENSITIVE) METRICS (IN %)

Classifier	Mean ±St.d.	Clip AUC	F-measure	Tag AUC
AdaBoost	MIREX	87.73±0.09	30.27±0.46	79.41±0.25
	CS Only	88.54±0.07	32.20±0.41	80.56±0.20
	ML Only	88.50±0.11	31.18±0.45	79.91±0.31
	CSML	**88.82±0.09**	**32.42±0.45**	**80.69±0.28**
SVM	MIREX	88.29±0.10	31.77±0.37	80.01±0.27
	CS Only	88.96±0.06	32.93±0.38	81.12±0.20
	ML Only	89.00±0.08	32.70±0.36	81.41±0.19
	CSML	**89.64±0.07**	**34.22±0.41**	**82.06±0.23**
Ensemble	MIREX	88.47±0.07	33.35±0.40	81.89±0.19
	CS Only	89.21±0.06	34.32±0.41	82.54±0.18
	ML Only	89.12±0.07	33.59±0.37	82.37±0.18
	CSML	**89.57±0.06**	**34.69±0.46**	**82.85±0.17**
RA*k*EL	ML Only	88.67±0.10	33.41±0.41	79.15±0.22
	CSML	**89.63±0.10**	**33.84±0.40**	**81.41±0.22**

Table IX compares the results of different methods in terms of the regular (cost-insensitive) evaluation metrics. Interestingly, the cost-sensitive stacking methods outperform their cost-insensitive binary classification counterparts in terms of the regular metrics (cf. AdaBoost-CS only versus AdaBoost-MIREX and AdaBoost-CSML versus AdaBoost-ML only). 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. Moreover, cost-sensitive RA*k*EL outperforms RA*k*EL in terms of all three metrics, and is better than

TABLE VII
AUDIO TAG ANNOTATION AND RETRIEVAL RESULTS OF DIFFERENT CLASSIFIERS AND ENSEMBLE METHODS ON THE MAJORMINER DATASET (IN %)

Mean	Clip AUC		F-measure		Tag AUC	
±St.d.	Without Seg.	With Seg.	Without Seg.	With Seg.	Without Seg.	With Seg.
AdaBoost	86.27±0.09	87.74±0.09	28.56±0.36	30.34±0.51	75.20±0.26	79.43±0.24
SVM	87.88±0.09	88.28±0.12	30.92±0.28	31.69±0.38	78.48±0.29	79.90±0.30
Probability Ensemble	87.88±0.07	**88.48±0.07**	31.63±0.37	32.96±0.39	78.94±0.30	81.08±0.20
Ranking Ensemble	76.26±0.12	78.14±0.10	32.11±0.32	**33.32±0.38**	79.97±0.22	**81.89±0.17**

AdaBoost-CSML and comparable to SVM-CSML.

Finally, we analyze the difference between the prediction results of the high count tags and low count tags in terms of the cost-insensitive false negative rate. The results are shown in Table X. The high count tags include tags whose counts are at least 6, and the low count tags are tags whose counts are 2. The results are extracted from one of the one hundred runs of three-fold cross-validation. Several observations can be drawn from Table X. First, there are much more low count tags than high count tags. Second, both Ensemble-MIREX and Ensemble-CSML have a significantly higher false negative rate on the low count tags than on the high count tags. The major reason could be that a low count tag captures less salient property, which is difficult to be recognized, of an audio clip than a high count tag. Third, Ensemble-CSML outperforms Ensemble-MIREX on both high count tags and low count tags. The reason has been explained earlier in this section.

TABLE X
COMPARISON OF PREDICTION RESULTS OF HIGH COUNT TAGS AND LOW COUNT TAGS IN TERMS OF FALSE NEGATIVE RATE

	High Count Tags	Low Count Tags
Number of Clip-Tag Pairs	686	4,418
Ensemble-MIREX	28.72	65.73
Ensemble-CSML	**26.38**	**64.42**

X. CONCLUSION

In this paper, we have presented our MIREX 2009 winning method for automatic audio tagging. The method combines both homogeneous segmentation and classifier ensemble techniques. After the competition, we have realized that the tag counts and the tag co-occurrences are important information that should be considered in automatic audio tagging. To exploit the tag count information, we have proposed formulating the audio tagging task as a cost-sensitive classification problem in order to minimize the misclassified tag counts. In addition, we have discussed several factors that affect the counts of tags assigned to an audio clip, and presented cost-sensitive versions of several regular evaluation metrics. To exploit the tag correlation information, we have proposed formulating the audio tagging task as a multi-label classification problem. To exploit the tag count and correlation information jointly, we have proposed formulating the audio tagging task as a cost-sensitive multi-label classification problem and extended two multi-label classification methods, namely, stacking and RA*k*EL, to their cost-sensitive versions to solve the problem. To the best of our knowledge, cost-sensitive multi-label classification has not been studied previously.

REFERENCES

[1] P. Lamere, "Social tagging and music information retrieval," *Journal of New Music Research*, vol. 37, no. 2, pp. 101–114, 2008.

[2] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green, "Automatic generation of social tags for music recommendation," in *Advances in Neural Information Processing Systems*, 2007.

[3] M. Hoffman, D. Blei, and P. Cook, "Easy as CBA: a simple probabilistic model for tagging music," in *Proc. Int. Society Music Information Retrieval Conference*, 2009.

[4] H.-Y. Lo, J.-C. Wang, and H.-M. Wang, "Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2010.

[5] S. Ness, A. Theocharis, L. G. Martins, and G. Tzanetakis, "Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs," in *Proc. ACM Int. Conf. Multimedia*, 2009.

[6] D. Tingle, Y. Kim, and D. Turnbull, "Exploring automatic music annotation with "acoustically-objective" tags," in *Proc. ACM Int. Conf. Multimedia Information Retrieval*, 2010.

[7] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, pp. 467–476, 2008.

[8] J. Foote and M. Cooper, "Media segmentation using self-similarity decomposition," in *Proc. SPIE Storage and Retrieval for Multimedia Databases*, 2003.

[9] M. I. Mandel and D. P. W. Ellis, "A web-based game for collecting music metadata," *Journal of New Music Research*, vol. 37, no. 2, pp. 151–165, 2008.

[10] X. Li, C. G. M. Snoek, and M. Worring, "Learning social tag relevance by neighbor voting," *IEEE Transactions on Multimedia*, vol. 11, pp. 1310–1322, 2009.

[11] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl, "Aggregate features and ADABOOST for music classification," *Machine Learning*, vol. 65, no. 2-3, pp. 473–484, 2006.

[12] M. I. Mandel and D. P. W. Ellis, "Multiple-instance learning for music information retrieval," in *Proc. Int. Conf. Music Information Retrieval*, 2007.

[13] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer, 2010.

[14] H.-T. Lin, "Cost-sensitive classification: Status and beyond," in *Workshop on Machine Learning Research in Taiwan: Challenges and Directions*, 2010.

[15] R. Miotto, L. Barrington, and G. Lanckriet, "Improving auto-tagging by modeling semantic co-occurrences," in *Proc. Int. Society Music Information Retrieval Conference*, 2010.

[16] L. Cao, J. Yu, J. Luo, and T. S. Huang, "Enhancing semantic and geographic annotation of web images via logistic canonical correlation regression," in *Proc. ACM Int. Conf. Multimedia*, 2009.

[17] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[18] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proc. European Conference on Machine Learning*, 2007.

[19] O. Lartillot and P. Toiviainen, "A matlab toolbox for musical feature extraction from audio," in *Proc. Int. Conf. Digital Audio Effects*, 2007.

[20] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–279, 1995.

[21] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[22] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang, “Cost-sensitive boosting for classification of imbalanced data,” *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

[23] J. Platt, “Probabilistic outputs for support vector machines and comparison to regularized likelihood methods,” in *Advances in Large Margin Classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 1998.

[24] H.-T. Lin, C.-J. Lin, and R. C. Weng, “A note on Platt’s probabilistic outputs for support vector machines,” *Machine Learning*, vol. 68, no. 3, pp. 267–276, 2007.

[25] B. Zadrozny, J. Langford, and N. Abe, “Cost-sensitive learning by cost-proportionate example weighting,” in *Proc. IEEE Int. Conf. Data Mining*, 2003.

[26] H.-T. Lin, “A simple cost-sensitive multiclass classification algorithm using one-versus-one comparisons,” National Taiwan University, Tech. Rep., December 2010.

[27] X. Hu, J. S. Downie, and A. Ehmann, “Lyric text mining in music mood classification,” in *Proc. Int. Society Music Information Retrieval Conference*, 2009.

Hung-Yi Lo received the B.S. degree in electronic engineering and the M.S. degree in computer science and information engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan.

In 2004, he joined the Speech, Language, and Music Processing Group, Institute of Information Science, Academia Sinica, Taiwan, as a Research Assistant. His research interests include data mining, machine learning, and music information retrieval. He participanted the NTU team to win the first place of the ACM KDD Cup in 2008 and 2010, and the third place in 2009.

Ju-Chiang Wang received the B.S. degree in computer science and information engineering from National Central University, Chung-Li, Taiwan, in 2004, and the M.S. degree in computer science and information engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, in 2007. He is currently pursuing the Ph.D. degree in the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan.

He joined the Speech, Language, and Music Processing Group, Institute of Information Science, Academia Sinica, Taiwan, as a Research Assistant in 2007. His research interests include music information retrieval, Mandarin singing voice processing and machine learning with applications on audio data.

Hsin-Min Wang (S‘92-M‘95-SM‘04) received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1989 and 1995, respectively.

In October 1995, he joined the Institute of Information Science, Academia Sinica, Taipei, as a Post-Doctoral Fellow. He was promoted to Assistant Research Fellow, Associate Research Fellow, and then Research Fellow, in 1996, 2002, and 2010, respectively. He was an Adjunct Associate Professor with the National Taipei University of Technology, Taipei, and National Chengchi University, Taipei. His major research interests include speech processing, natural language processing, multimedia information retrieval, and pattern recognition.

Dr. Wang was a recipient of the Chinese Institute of Engineers Technical Paper Award in 1995. He was a board member, chair of academic council, and secretary-general of the Association for Computational Linguistics and Chinese Language Processing (ACLCLP). He currently serves as a standing board member of ACLCLP and as an editorial board member of the International Journal of Computational Linguistics and Chinese Language Processing. He is a life member of ACLCLP and Institute of Information and Computing Machinery, and is a member of International Speech Communication Association.

Shou-De Lin holds a BS in EE from National Taiwan University, an MS-EE from the University of Michigan, and an MS in Computational Linguistics and PhD in Computer Science both from the University of Southern California. In 2007, he joined the CSIE Department of National Taiwan University as an assistant professor. He leads the Machine Discovery and Social Network Mining Lab in NTU. Before joining NTU, he was a post-doctoral research fellow at the Los Alamos National Lab. Prof. Lin’s research includes the areas of knowledge discovery and data mining, social network analysis, natural language processing and machine learning. His international recognition includes the best paper award in IEEE Web Intelligent conference 2003, Google Research Award in 2007, and Microsoft research award in 2008. He leads or co-leads the NTU team to win the first place of the ACM KDD Cup in 2008 and 2010, and the third place in 2009.