COST-SENSITIVE STACKING FOR AUDIO TAG ANNOTATION AND RETRIEVAL

Hung-Yi Lo[1,2], *Ju-Chiang Wang*[1], *Hsin-Min Wang*[1], *Shou-De Lin*[2]

[1]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[2]Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
{hungyi, asriver, whm}@iis.sinica.edu.tw, sdlin@csie.ntu.edu.tw

ABSTRACT

Audio tags correspond to keywords that people use to describe different aspects of a music clip, such as the genre, mood, and instrumentation. 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 is another useful information 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. The results of audio tag annotation and retrieval experiments 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

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 [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 [2–5], 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 area under the receiver operating characteristic curve (AUC) given tag and F-measure [3]. This paper aims to improve our MIREX winning method by 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.

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[2], a web-based game for collecting music tags. 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 *most salient property* of the audio clip. The count also reflects the popularity of the tag, song, artist, and album. In addition, a tag with a small count may contain noisy information, which would affect the training of the tag classifier. The cues indicating noisy labeling are not considered in previous methods. 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 auto-

This work was partially supported 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.

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

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

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)

matic 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 less likely to be also annotated with “guitar” than “drum”. However, previous research [2, 3] 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 lose 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. 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 a cost-sensitive stacking method to solve it. To the best of our knowledge, cost-sensitive multi-label classification has not been studied previously.

The remainder of this paper is organized as follows. In Section 2, we give an overview of our audio tag annotation and retrieval system. Then, we present the proposed cost-sensitive multi-label classification method in Section 3. We discuss the results of our experiments in Section 4. Finally, Section 5 contains some concluding remarks.

2. 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. The classification models will be detailed in the next section.

3. COST-SENSITIVE MULTI-LABEL LEARNING

We first introduce the concept of multi-label classification. Let $\boldsymbol{x} \in \mathbb{R}^d$, which is a d-dimensional input space, and $\boldsymbol{y} \subseteq \mathcal{L} = \{1, 2, ..., K\}$, which is a finite set of K possible labels. 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 a set of proper labels for an unseen sample $\boldsymbol{x}$. The set of labels can be represented by $\boldsymbol{y} = (y_1, y_2, ..., y_K) \in \{1, -1\}^K$.

Cost-sensitive multi-label (CSML) 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 an existing multi-label learning algorithm, namely stacking, to its cost-sensitive version to solve the CSML problem.

3.1. Cost-sensitive Classification

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. To train the cost-sensitive SVM classifier for the j-th tag, the objective function is formulated as follows:

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

where $\boldsymbol{w}$ is the parameter to 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_{ij} is associated with a corresponding training error term ξ_i.

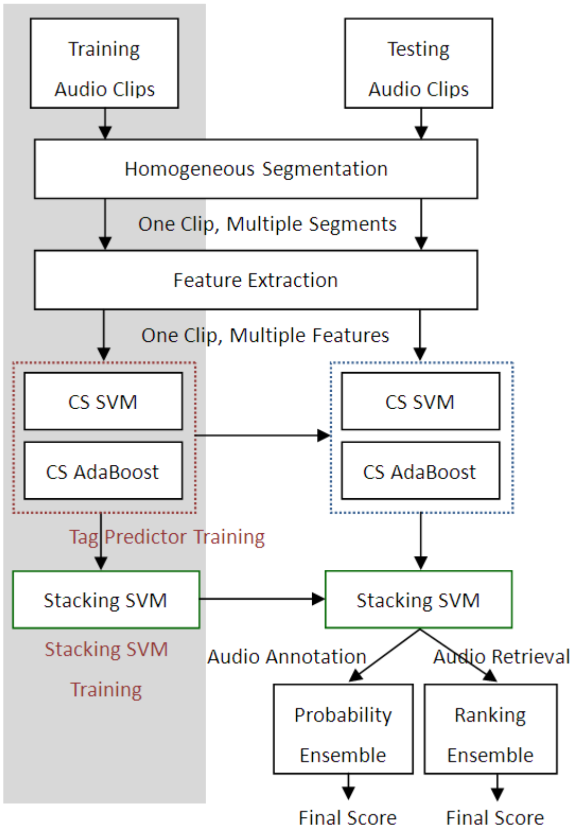

Fig. 1. The work flow of the proposed audio tag annotation and retrieval system.

AdaBoost finds a highly accurate classifier by combining several base classifiers, even though each of them is only moderately accurate. Cost-sensitive AdaBoost [7] 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. When training the j-th tag classifier, in each iteration, the weight vector D_t is updated by

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t c_{ij} y_{ij} h_t(\boldsymbol{x}_i))}{Z_t}, \qquad (2)$$

where $h_t(\boldsymbol{x}_i)$ is the prediction score of the base classifier h_t for instance $\boldsymbol{x}_i$; 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.

3.2. 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_{ij} of the stacking classifiers:

$$h_i(\boldsymbol{z}) = \sum_{j=1}^{K} w_{ij} z_j, \qquad (3)$$

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

Inspired by the idea of stacking, we improve our MIREX 2009 classifier ensemble by using cost-sensitive stacking. 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 two sets of binary tag classifiers. Finally, the stacked 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. EXPERIMENTS

The experiments follow our previous setup of the MIREX 2009 extended experiments reported in [3]. 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 [8]. Then, using MIRToolbox 1.1[3], 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.

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

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

Table 2. Audio Tag Annotation and Retrieval Results of Cost-sensitive Multi-label Classification Methods (in %)

	Mean ±St.d.	Clip AUC	F-measure	Tag AUC
Ada-Boost	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**
Ens-emble	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**

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

Our experiment results in terms of the metrics corresponding to the audio tag retrieval task and the audio tag annotation task are summarized in Table 2. Because the cross-validation split used in MIREX 2009 is not available, we perform three-fold cross-validation one hundred times and calculate the mean and standard deviation of the results to reduce the variance of different cross-validation splits. 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. The Ensemble methods use probability ensemble to generate Clip AUC and ranking ensemble to generate F-measure and Tag AUC. The ML only methods employ stacking and the CSML methods employ cost-sensitive stacking.

The results demonstrate the effectiveness of CSML learning. The improvement in F-measure is the most significant: 2.15% for AdaBoost-CSML versus AdaBoost-MIREX, 2.45% for SVM-CSML versus SVM-MIREX, and 1.34% for Ensemble-CSML versus Ensemble-MIREX. 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). 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.

5. CONCLUSION

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. 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 a multi-label classification method, namely stacking, to its cost-sensitive version to solve the problem. To the best of our knowledge, cost-sensitive multi-label classification has not been studied previously.

6. REFERENCES

[1] Paul 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 *NIPS*, 2007.

[3] Hung-Yi Lo, Ju-Chiang Wang, and Hsin-Min Wang, "Homogeneous segmentation and classifier ensemble for audio tag annotation and retrieval," in *ICME*, 2010.

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

[5] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, pp. 467–476, 2008.

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

[7] Yanmin Sun, Mohamed S. Kamel, Andrew K. C. Wong, and Yang Wang, "Cost-sensitive boosting for classification of imbalanced data," *Pattern Recognition*, vol. 40, no. 12, pp. 3358–3378, 2007.

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