AUDIO CLASSIFICATION USING SEMANTIC TRANSFORMATION AND CLASSIFIER ENSEMBLE

Ju-Chiang Wang[*†], *Hung-Yi Lo*[†], *Shyh-Kang Jeng*[*] *and Hsin-Min Wang*[†]

[*]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan
[†]Institute of Information Science, Academia Sinica, Taipei, Taiwan
E-mail: asriver@iis.sinica.edu.tw, hungyi@iis.sinica.edu.tw,
skjeng@cc.ee.ntu.edu.tw, whm@iis.sinica.edu.tw

ABSTRACT

This paper presents our winning audio classification system in MIREX 2010. Our system is implemented as follows. First, in the training phase, the frame-based 70-dimensional feature vectors are extracted from a training audio clip by MIRToolbox. Next, the Posterior Weighted Bernoulli Mixture Model (PWBMM) is applied to transform the frame-decomposed feature vectors of the training song into a fixed-dimensional semantic vector representation based on the pre-defined music tags; this procedure is called Semantic Transformation. Finally, for each class, the semantic vectors of associated training clips are used to train an ensemble classifier consisting of SVM and AdaBoost classifiers. In the classification phase, a testing audio clip is first represented by a semantic vector, and then the class with the highest score is selected as the final output. Our system was ranked first out of 36 submissions in the MIREX 2010 audio mood classification task.

1. INTRODUCTION

Automatic music classification is a very important topic in the music information retrieval (MIR) field. It was first addressed by Tzanetakis et al., who worked on automatic musical genre classification of audio signals in 2001 [1]. After ten years of development, many kinds of audio classification datasets have been created with category definitions and class labels corresponding to a set of audio examples. In addition, many approaches have been proposed for classifying music data according to genre [1, 2], mood [3, 4], or artists [5, 6]. Music Information Retrieval Evaluation eXchange (MIREX), an annul MIR algorithm competition held jointly with ISMIR, started to evaluate audio classification from 2005.

In the audio classification field, fixed numbers of categories or classes are usually pre-defined by experts for different application tasks. In general, these categories or classes should be definite and as mutually exclusive as possible. However, when most people listen to a song they have never heard before, they usually have certain musical impressions in their minds, although they may not be able to name the exact musical category of the song. These musical impressions inspired by direct auditory cues can be described by some general words, such as exciting, noisy, fast, male vocal, drum, and guitar. We believe that the co-occurrences of the musical impressions or concepts may indicate the membership of a song in a specific audio class. Therefore, in this study, we will explore the relationship between the general tag words and the specific categories.

Since people tend to mentally tag a piece of music with specific words when they listen to it, music tags are a natural way to describe the general musical concepts. The tags can include different types of musical information, such as genre, mood, and instrumentation. Therefore, we believe that the knowledge of pre-generated music tags in a music dataset can help the classification of another music dataset. In other words, we can train a music tagging system to recognize musical concepts of a song in terms of semantic tags first, and then the music classification system can classify the song into specific classes based on the semantic representation.

Figure 1 shows an overview of our music classification system. There are two layers in our system, i.e., semantic transformation (ST) and ensemble classification (EC). In the training phase of the ST layer, we first extract audio features with respect to various types of musical characteristics, including dynamics, spectral, timbre, and tonal features, from the training audio clips. Next, we apply the Posterior Weighted Bernoulli Mixture Model (PWBMM) [7] to automatically tag the clips. The PWBMM performed very well in terms of the tag-based area under the receiver operating characteristic curve (AUC-ROC) in the MIREX 2010 audio tag classification task [8]. The AUC-ROC of the tag affinity output is an important way to evaluate the correct tendency of the tagging prediction; therefore, we have proper confidence in applying the PWBMM in the music tagging step in our system. The PWBMM is trained on the MajorMiner dataset clawed from the website of the MajorMiner [1] music tagging game. The dataset contains 2,472 ten-second audio clips and their associated tags. As shown in Table 1, we select 45 tags to define the semantic space. In other words, a

[1] http://majorminer.org/

song is transformed into a 45-dimensional semantic vector over the pre-defined tags by ST based on the tagging procedure. In the MajorMiner dataset, the counts of a tag given to a music clip ranges from 2 to 12. These counts are also modeled by PWBMM and have been shown to facilitate the performance of music tag annotation [7]. In the training phase of the EC layer, for each class, the associated training audio clips, each represented by a 45-dimensional semantic vector, are used to train an ensemble classifier, consisting of support vector machine (SVM) and AdaBoost classifiers. In the final classification phase, given a testing audio clip, the class with the highest output score is assigned to it.

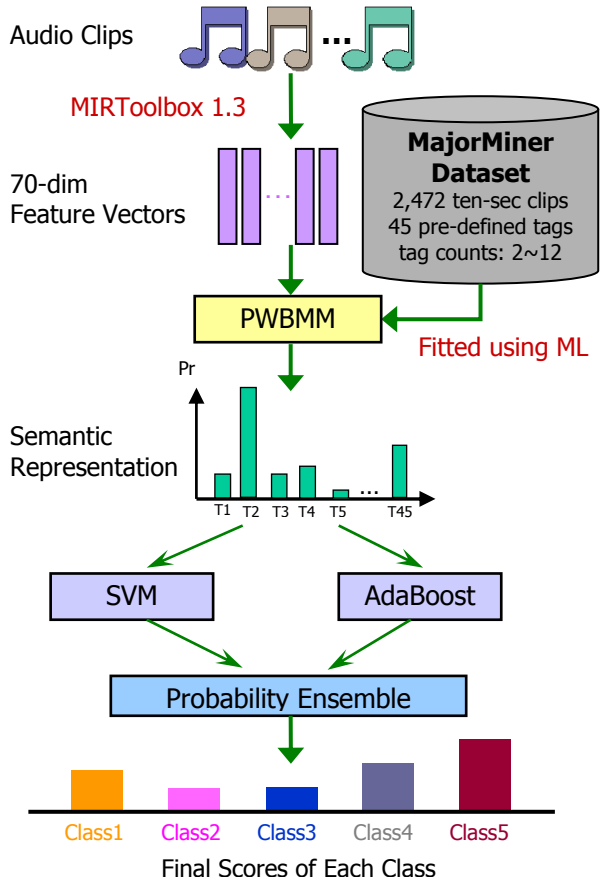

Figure 1. The flowchart of our audio classification system.

The remainder of this paper is organized as follows. In Section 2, we describe the music features used in this work. In Section 3, we present how to apply PWBMM for music semantic representation, and in Section 4, we present our ensemble classification method. We introduce the MIREX 2010 audio train/test: mood classification task and discuss the results in Section 5. Finally, the conclusion is given in Section 6.

Table 1. The 45 tags used in our music classification system.

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

2. MUSIC FEATURE EXTRACTION

We use MIRToolbox 1.3[2] for music feature extraction [9]. As shown in Table 2, four types of features are used in our system, including dynamics, spectral features, timbre, and tonal features. To ensure the alignment and prevent the mismatch of different features in a vector, all the features are extracted from the same fixed-size short-time frame. Given a song, a sequence of 70-dimensional feature vectors is extracted with 50ms frame size and 0.5 hop shift.

Table 2. The music features used in the 70-dimensional frame-based vector.

Types	Feature Description	Dim
dynamics	rms	1
spectral	centroid	1
	spread	1
	skewness	1
	kurtosis	1
	entropy	1
	flatness	1
	rolloff at 85%	1
	rolloff at 95%	1
	brightness	1
	roughness	1
	irregularity	1
timbre	zero crossing rate	1
	spectral flux	1
	MFCC	13
	delta MFCC	13
	delta-delta MFCC	13
tonal	key clarity	1
	key mode possibility	1
	HCDF	1
	chroma peak	1
	chroma centroid	1
	chroma	12

[2] http://www.jyu.fi/music/coe/materials/mirtoolbox

3. POSTERIOR WEIGHTED BERNOULLI MIXTURE MODEL

The PWBMM-based music tagging system contains two steps. First, it converts the frame-based feature vectors of a song into a fixed-dimensional vector (in a Gaussian Mixture Model (GMM) posterior representation). Then, the Bernoulli Mixture Model (BMM) [10] predicts the scores over 45 music tags for the song.

3.1. GMM Posterior Representation

Before training the GMM, the feature vectors from all training audio clips are normalized to have a mean of 0 and standard deviation of 1 in each dimension. Then, the GMM is fitted by using the expectation and maximization (EM) algorithm. The generation of the GMM posterior representation can be viewed as a process of soft tokenization from a music background model. We address a "latent music class" as a latent variable $z_k \in \{z_1, z_2, \ldots, z_K\}$ corresponding to the k-th Gaussian component with mixture weight π_k, mean vector $\boldsymbol{\mu}_k$, and covariance matrix $\boldsymbol{\Sigma}_k$ in the GMM. With the GMM, we can describe how likely a given feature vector $\mathbf{x}$ belongs to a "latent music class" z_k by the posterior probability of the latent music class:

$$p(z_k = 1|\mathbf{x}) = \frac{\pi_k N(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{i=1}^{K} \pi_i N(\mathbf{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}. \quad (1)$$

Given a song s_j, by assuming that each frame contributes equally to the song, the posterior probability of a certain latent music class can be computed by

$$p(z_k = 1 \mid s_j) = \frac{1}{N_j} \sum_{n=1}^{N_j} p(z_k = 1|\mathbf{x}_{jn}), \quad (2)$$

where $\mathbf{x}_{jn}$ is the feature vector of the n-th frame of song s_j and N_j is the number of frames in song s_j.

3.2. Bernoulli Mixture Model

Assume that we have a training music corpus with J audio clips, each denoted as s_j, j=1,…,J, and with associated tag counts c_{jw}, w=1,…,W. The tag counts are positive integers indicating the number of times that tag t_w has been assigned to clip s_j. The binary random variable $\boldsymbol{y}$, with $y_{jw} \in \{0,1\}$, represents the event of tag t_w applying to song s_j.

3.2.1. Generative Process

The generative process of BMM has two steps. First, the probability of the latent class z_{kw}, k=1,…,K, is chosen from song s_j's class weight vector $\boldsymbol{\theta}_j$:

$$p(z_{kw} = 1 \mid \boldsymbol{\theta}_j) = \theta_{jk}, \quad (3)$$

where θ_{jk} is the weight of the k-th latent class. Second, a case of the discrete variable y_{jw} is selected based on the following conditional probabilities:

$$\begin{aligned} p(y_{jw} = 1 \mid z_{kw} = 1, \boldsymbol{\beta}) &= \beta_{kw} \\ p(y_{jw} = 0 \mid z_{kw} = 1, \boldsymbol{\beta}) &= 1 - \beta_{kw} \end{aligned}. \quad (4)$$

The conditional probability that models the probability of clip s_j having tag t_w is a Bernoulli distribution with input discrete variable y_{jw} and parameter $\boldsymbol{\beta}$ for the k-th class z_{kw}.

The complete joint distribution over $\boldsymbol{y}$ and $\boldsymbol{z}$ is described with model parameter $\boldsymbol{\beta}$ and weight matrix $\boldsymbol{\Theta}$, where its row vector is $\boldsymbol{\theta}_j$ of clip s_j:

$$p(\boldsymbol{y}, \boldsymbol{z} \mid \boldsymbol{\beta}, \boldsymbol{\Theta}) = \prod_{j=1}^{J} p(\boldsymbol{y}, z_{kw} = 1 \mid \boldsymbol{\beta}, \boldsymbol{\theta}_j) = \prod_{j=1}^{J} \prod_{w=1}^{W} \theta_{jk}^{z_{kw}} \beta_{kw}. \quad (5)$$

The marginal log likelihood of the music corpus can be expressed as:

$$\log p(\boldsymbol{y} \mid \boldsymbol{\beta}, \boldsymbol{\Theta}) = \sum_{j=1}^{J} \sum_{w=1}^{W} \log \left\{ \sum_{k=1}^{K} \theta_{jk}^{z_{kw}} \beta_{kw} \right\}. \quad (6)$$

3.2.2. Model Inference by the EM Algorithm

The BMM can be fitted with respect to parameter $\boldsymbol{\beta}$ and weight matrix $\boldsymbol{\Theta}$ by maximum-likelihood (ML) estimation. By linking the latent class of BMM with the "latent music class" of GMM described in Section 3.1, the posterior probability in Eq. (2) can be viewed as the class weight, i.e., $\theta_{jk} = p(z_k{=}1|s_j)$. Therefore, we only need to estimate $\boldsymbol{\beta}$, which corresponds to the probability that a latent music class occurs. We apply the EM algorithm to maximize the corpus-level log-likelihood in Eq. (6) in the presence of latent variable $\boldsymbol{z}$.

In the E-step, given the clip-level weight matrix $\boldsymbol{\Theta}$ and the model parameter $\boldsymbol{\beta}$, the posterior probability of each latent variable z_{kw} can be computed by

$$\begin{aligned} \gamma_j(z_{kw}) &= p(z_{kw} = 1 \mid \boldsymbol{\beta}, \boldsymbol{\Theta}, \boldsymbol{y}) \\ &= \frac{p(y_{jw} \mid z_{kw} = 1, \boldsymbol{\beta}) p(z_{kw} = 1 \mid \boldsymbol{\theta}_j)}{p(y_{jw} \mid \boldsymbol{\beta}, \boldsymbol{\theta}_j)} \\ &= \begin{cases} \frac{\theta_{jk}\beta_{kw}}{\sum_{i=1}^{K} \theta_{jk}\beta_{kw}} & \text{for } y_{jw} = 1 \\ \frac{\theta_{jk}(1-\beta_{kw})}{\sum_{i=1}^{K} \theta_{jk}(1-\beta_{kw})} & \text{for } y_{jw} = 0. \end{cases} \end{aligned} \quad (7)$$

In the M-step, the update rule for β_{kw} is as follows,

$$\beta_{kw} \leftarrow \frac{\sum_j \gamma_j(z_{kw}) y_{jw}}{\sum_j \gamma_j(z_{kw})}. \quad (8)$$

From the tag counts of the music corpus, we know that there exist different levels of relationship between a clip and a tag. If clip s_j has a more-than-one tag count c_{jw} for tag t_w, we can make song s_j contribute to β_{kw} c_{jw} times rather than only once in each iteration of EM. This leads to a new update rule for β_{kw}:

$$\beta_{kw} \leftarrow \frac{\sum_j c_{jw} \gamma_j(z_{kw}) y_{jw}}{\sum_{j, y_{jw}=1} c_{jw} \gamma_j(z_{kw}) + \sum_{j, y_{jw}=0} \gamma_j(z_{kw})}. \quad (9)$$

3.2.3. Semantic Transformation with PWBMM

The w-th component of the semantic vector $\mathbf{v}$ of a given clip s is computed as the conditional probability of $y_w = 1$ given $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$:

$$p(y_w = 1 | \boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \theta_k \, p(y_w = 1 | \beta_{kw}) = \sum_{k=1}^{K} \theta_k \beta_{kw}. \quad (10)$$

For the ensemble classification layer, given an audio clip s_m, m=1,2,…,M, its semantic representation is generated in the same way. First, a sequence of music feature vectors is extracted from s_m. Second, the vector sequence is transformed into a fixed dimensional posterior weight vector $\boldsymbol{\theta}_m$ via Eq. (2). Third, the weight vector $\boldsymbol{\theta}_m$ is transformed into a fixed dimensional semantic vector $\mathbf{v}_m$ via Eq. (10).

4. THE ENSEMBLE CLASSIFICATION METHOD

Assume that we have G classes for the audio classification task, and all the classes are independent. We can train G binary ensemble classifiers, denoted as $\mathbf{C}_g$, $g = 1, 2, \ldots, G$, for each class. Each ensemble classifier $\mathbf{C}_g$ calculates a final score by combining the outputs of two sub-classifiers: SVM and AdaBoost.

4.1. 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. The large margin implies good generalization ability in theory. In this work, we exploited a linear SVM classifier $f(\mathbf{v})$ of the following form:

$$f(\mathbf{v}) = \sum_{w=1}^{W} \lambda_w v_w + b, \quad (11)$$

where v_w is the w-th component of the semantic vector $\mathbf{v}$ of a testing clip; λ_w and b are parameters to be trained from ($\mathbf{v}_m$, l_{mg}), m=1,…,M, where $\mathbf{v}_m$ is the semantic vector of the m-th training clip and $l_{mg} \in \{1, 0\}$ is the g-th class label of the m-th training clip; and W is the dimension of the semantic vector. The advantage of linear SVM is training efficiency. Certain recent literature has shown that it has comparable prediction performance compared to non-linear SVM. A single cost parameter is determined by using cross-validation.

4.2. AdaBoost

Boosting is a method of finding a highly accurate classifier by combining several base classifiers, even though each of them is only moderately accurate. We use decision stumps as the base learner. The decision function of the boosting classifier takes the following form:

$$g(\mathbf{v}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{v}), \quad (12)$$

where α_t is set as suggested in [5]. The model selection procedure can be done efficiently as we can iteratively increase the number of base learners and stop when the generalization ability with respect to the validation set does not improve.

4.3. Calibrated Probability Scores and Probability Ensemble

The ensemble classifier averages the scores of the two sub-classifiers, i.e., SVM and AdaBoost. However, since the sub-classifiers for different classes are trained independently, their raw scores are not comparable. Therefore, we transform the raw scores of the sub-classifiers into probability scores with a sigmoid function [7]:

$$\Pr(l = 1 | \mathbf{v}) \approx \frac{1}{1 + \exp(Af + B)}, \quad (13)$$

where f is the raw score of a sub-classifier, and A and B are learned by solving a regularized maximum likelihood problem [8]. As the sub-classifier output has been calibrated into a probability score, a classifier ensemble for a specific class is formed by averaging the probability scores of associated SVM and AdaBoost sub-classifiers, and the probability scores of classifiers for different classes become comparable. The class with the highest output score is assigned to a testing music clip.

4.4. Cross-Validation

We first perform inner cross-validation on the training

set to determine the cost parameter *C* of linear SVM and the number of base learners in AdaBoost. Then, we re-train the classifiers with the complete training set and the selected parameters. We use the AUC-ROC as the model selection criterion.

5. MIREX 2010 AUDIO TRAIN/TEST: MUSIC MOOD CLASSIFICATION

We submitted our audio classification system described above to the MIREX 2010 Audio Train/Test tasks. Due to some unknown reasons, only the evaluation results on the music mood dataset were reported (this also happens to some other teams), although we believe that our system is dedicated to adapt to any kinds of audio classification datasets. In the following discussions, this system is denoted as WLJW2. We also submitted a simple system (WLJW1) as a baseline system. In WLJW1, the representation of an audio clip is the mean vector of all frame-based feature vectors of the clip, and a simple quadratic classifier [15] for each class is trained.

5.1. The Music Mood Dataset

The music mood dataset [4] was first used in MIREX 2007. There are 600 30-second audio clips in 22,050Hz mono wave format selected from the APM collection[3]. The corresponding five mood categories, each contains 120 clips, are shown in Table 3. The mood class of an audio clip is labeled by human judges using the Evalutron 6000 system [16].

Table 3. The five mood categories and their components.

Class	Mood Components
1	passionate, rousing, confident, boisterous, rowdy
2	rollicking, cheerful, fun, sweet, amiable/good natured
3	literate, poignant, wistful, bittersweet, autumnal, brooding
4	humorous, silly, campy, quirky, whimsical, witty, wry
5	aggressive, fiery, tense/anxious, intense, volatile, visceral

5.2. Evaluation Results

MIREX uses three-fold cross-validation to evaluate the systems submitted. In each fold, one subset is selected as the test set and the remaining two subsets serve as the training set. The performance is summarized in Table 4 [17]. The summary accuracy is the average accuracy of the three folds. The bold values represent the best performance in each evaluation metric.

Table 4. The performance of all submissions on the music mood dataset.

Submission Code	Summary Accuracy	Accuracy per Testing Fold		
		0	1	2
WLJW1	0.5383	0.590	0.500	0.525
WLJW2	**0.6417**	**0.735**	0.595	0.595
BMPE2	0.5467	0.585	0.505	0.550
BRPC1	0.5867	0.645	0.575	0.540
BRPC2	0.5900	0.695	0.550	0.525
CH1	0.6300	0.705	0.615	0.570
CH2	0.6300	0.725	0.605	0.560
CH3	0.6350	0.710	**0.640**	0.555
CH4	0.6267	0.710	0.615	0.555
FCY1	0.6017	0.710	0.540	0.555
FCY2	0.5950	0.685	0.550	0.550
FE1	0.6083	0.690	0.555	0.580
GP1	0.6317	0.695	0.565	**0.635**
GR1	0.6067	0.685	0.570	0.565
HE1	0.5417	0.580	0.520	0.525
JR1	0.4633	0.480	0.435	0.475
JR2	0.5117	0.535	0.520	0.480
JR3	0.4683	0.475	0.475	0.455
JR4	0.5117	0.560	0.510	0.465
MBP1	0.5400	0.585	0.530	0.505
MP2	0.3617	0.200	0.385	0.500
MW1	0.5400	0.600	0.520	0.500
RJ1	0.5483	0.570	0.555	0.520
RJ2	0.5017	0.505	0.495	0.505
RK1	0.5483	0.595	0.520	0.530
RK2	0.4767	0.515	0.450	0.465
RRS1	0.6167	0.695	0.595	0.560
SSPK1	0.6383	0.665	0.630	0.620
TN1	0.5550	0.650	0.515	0.500
TN2	0.4858	0.540	0.430	0.480
TN4	0.5750	0.645	0.540	0.540
TS1	0.6100	0.705	0.575	0.550
WLB1	0.5550	0.605	0.535	0.525
WLB2	0.5767	0.625	0.550	0.555
WLB3	0.6300	0.690	0.615	0.585
WLB4	0.6300	0.705	0.600	0.585

It is clear that our system WLJW2 is ranked first out of 36 submissions in terms of summary accuracy. The summary accuracy of WLJW2 is 10.34% higher than that of our baseline system WLJW1. The results demonstrate that semantic transformation and classifier ensemble indeed enhance the audio classification performance. MIREX has also performed significance tests, and the results are shown in Figure 2. Figure 3 shows the overall class-pairs confusion matrix of WLJW2. According to the confusion matrix, our system reveals high confidence in classes 3 and 5, and the accuracies are 83.33% and 88.33%, respectively.

[3] http://www.apmmusic.com/

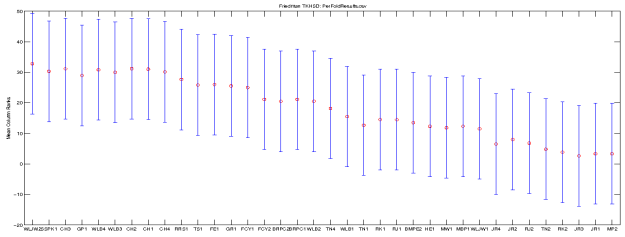

Figure 2. The significance tests on accuracy per fold by Friedman's ANOVA w/ Tukey Kramer HSD [17].

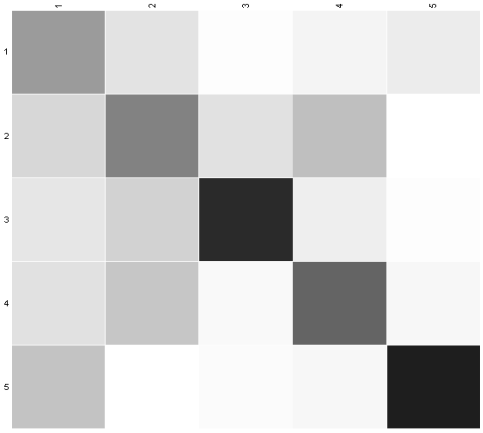

Figure 3. The overall confusion matrix of WLJW2.

6. CONCLUSIONS

In this paper, we have presented a music classification system integrating two layers of prediction based on semantic transformation and ensemble classification. The semantic transformation provides a musically conceptual representation, which matches human auditory sense to some extent, to a given audio clip. The robust ensemble classifier facilitates the final classification step. The results of MIREX evaluation tasks have shown that our system achieves very good performance compared to other systems.

7. ACKNOWLEDGEMENTS

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

8 REFERENCES

[1] G. Tzanetakis, G. Essl, and P. Cook, “Automatic Musical Genre Classification of Audio Signals,” *ISMIR*, 2001.

[2] T. Li, M. Ogihara, and Q. Li, “A Comparative Study on Content-Based Music Genre Classification,” *ACM SIGIR*, 2003.

[3] D. Liu, L. Lu, and H.-J. Zhang, “Automatic Mood Detection from Acoustic Music Data,” *ISMIR*, 2003.

[4] X. Hu, J. S. Downie, C. Laurier, M. Bay, and A. F. Ehmann, “The 2007 MIREX Audio Mood Classification Task: Lessons Learned,” *ISMIR*, 2008.

[5] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence, "The Quest for Ground Truth in Musical Artist Similarity," *ISMIR*, 2002.

[6] T. Li and M. Ogihara, “Music Artist Style Identification by Semisupervised Learning from both Lyrics and Content,” *ACM MM*, 2004.

[7] J.-C. Wang, H.-S. Lee, S.-K. Jeng, and H.-M. Wang, “Posterior Weighted Bernoulli Mixture Model for Music Tag Annotation and Retrieval,” *APSIPA ASC*, 2010.

[8] MIREX 2010 Results: Audio Tag Affinity Estimation, Submission Code: WLJW3, Name: Adaptive PWBMM, http://nema.lis.illinois.edu/nema_out/mirex2010/results/atg/subtask2_report/aff/

[9] O. Lartillot and P. Toiviainen, “A Matlab Toolbox for Musical Feature Extraction from Audio,” *DAFx*, 2007.

[10] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[11] 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.

[12] H.-Y. Lo, J.-C. Wang, and H.-M. Wang, “Homogeneous Segmentation and Classifier Ensemble for Audio Tag Annotation and Retrieval,” *ICME*, 2010.

[13] J. Platt, “Probabilistic Outputs for Support Vector Machines and Comparison to Regularized Likelihood Methods,” *Advances in Large Margin Classifiers*, Cambridge, MA.

[14] 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.

[15] W. J. Krzanowski, *Principles of Multivariate Analysis: A User's Perspective*, New York: Oxford University Press, 1988.

[16] A. A. Gruzd, J. S. Downie, M. C. Jones, and J. H. Lee, “Evalutron 6000: Collecting Music Relevance Judgments,” *ACM JCDL*, 2007.

[17] MIREX 2010 Results: Audio Mood Classification, http://nema.lis.illinois.edu/nema_out/9b11a5c8-9fcf-4029-95eb-51ed561cfb5f/results/evaluation/index.html