Discriminative training of GMM parameters for audio scene classification and audio tagging

Sungrack Yun, Sungwoong Kim, Sunkuk Moon, Juncheol Cho, Taesu Kim

Qualcomm Research
119 Nonhyun-dong, Gangnam-gu
Seoul, 135-820, KOREA
{sungrack, swkim, sunkukm, juncheol, taesu}@qti.qualcomm.com

ABSTRACT

This report describes the algorithm for audio scene classification and audio tagging and the result for DCASE 2016 challenge data. We propose a discriminative training algorithm to improve the baseline GMM performance. The algorithm updates the baseline GMM parameters by maximizing the margin between classes to improve discriminative performance. For Task1, we use a hierarchical classifier to maximize discriminative performance, and achieve 84% accuracy for given cross validation data. For Task4, we apply binary classifier for each label, and achieve 16.71% EER for given cross validation data.

Index Terms— audio scene classification, audio tagging, multi-label classification, discriminative training, Gaussian mixture model (GMM)

1. INTRODUCTION

For human, auditory perception plays a critical role to aware environments and interact with surroundings. Since speech is the most important auditory input, automatic speech recognition has attracted many researchers for several decades [1], and has been significantly improved by recent advances in machine learning technology [2]. Another important auditory information is music, because people love music, and there are a lot of applications and related industry. Music and speech discrimination is used for unified audio codec that can encode and decode both speech and music [3]. Music transcription is one of popular research topic in music information retrieval [4]. Identifying music title and artist from a short snippet of a microphone input is one of popular application in smart phone [5]. However, other types of auditory input is less investigated and a relatively new area. Recognizing various types of sound may bring many potential usages. For example, discriminating car, bus, subway, and street sound may help mobile map application to be smarter to suggest the corresponding route. If home robot or smart device can recognize audio event such as window breaking, fire alarm, and baby crying sound, it can notify the user when the event happens. The challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) is initiated to stimulate research in classifying and detecting such audio types. In recent advances in machine learning, deep neural network (DNN) based algorithms outperform many other conventional algorithms with the help of large database. It is expected that most of participants adopt variants of DNN algorithms. However, we have used GMM model which can be translated to single layered model, and propose an algorithm to update the model parameter to maximize the discrimination margin. There are two reasons why we have used single layered model for the tasks. First, DNN based algorithm did not outperform the proposed algorithm by a meaningful margin when we use the same feature vectors and the same size of the dataset. In our internal experiments, the performance of DNN based algorithms such as convolutive neural network (CNN) and recurrent neural networks (RNN) increases as the dataset size increases by data augmentation. But, our focus is to investigate the effectiveness of the criterion that we propose for discriminative training. This criterion can be further extended for DNN based algorithms. Secondly, we try to make the algorithm as simple as possible, because the potential applications of the system may have restrictions in computation or power. We apply the discriminative criterion, and achieve 84% accuracy for Task1 cross validation and 16.71% EER for Task4 cross validation without increment of computation complexity compared to the baseline system.

2. METHOD

2.1. Features

Given the Task1 wave files with the format of 44.1kHz, stereo, and 24bit, we first convert them into the wave files with the format of 44.1kHz, mono, and 16bit. Then, the Mel-frequency cepstral coefficient (MFCC) feature vectors of 60 dimensions including delta and acceleration coefficients are extracted. For Task1, we use given 16kHz sampled mono wave files to extract MFCC feature vectors of 60 dimensions including delta and acceleration coefficients.

2.2. Model

The Gaussian Mixture Model (GMM) is used to model each scene, and we obtain the classification label y^* of test input $X = \{x_1, \dots, x_T\}$ by

$$y^* = \underset{y \in Y}{\operatorname{argmax}} F(X, y; \theta)$$

where Y is the set of M scene labels, and θ is the GMM parameter set. From each speech frame, we obtain the feature vector $x_t, 1 \leq t \leq T$.The discriminant function $F(X, y; \theta)$ can be modeled using the conditional distribution $log p_\theta(y|X)$, and this can be expressed as

$$\begin{aligned} F(X, y; \theta) &= log p_\theta(y|X) \\ &= log p_\theta(X|y) p(y) \end{aligned}$$

where $p(y)$ is the prior probability of the classification label, and we assume equal prior probability, i.e. $p(y) = 1/M$, for all $y \in Y$. When using the K-mixture GMM with diagonal covariance, the probability $p_\theta(x_t|y)$ can be expressed as

$$p_\theta(x_t|y) = \sum_{k=1}^{K} w_k N(x_t; \mu_k, \sigma_k)$$

where w_k, μ_k, and σ_k are Gaussian mixture weight, mean vector, and variance of k-th mixture component.

With the assumption that x_t is independent and identically distributed, the discriminant function can be expressed as

$$\begin{aligned} F(X, y; \theta) &= log p_\theta(X|y) p(y) \\ &= log \left[\prod_{t=1}^{T} p_\theta(x_t|y) \cdot \frac{1}{M} \right] \\ &= log \left[\prod_{t=1}^{T} \sum_{k=1}^{K} w_k N(x_t; \mu_k, \sigma_k) \cdot \frac{1}{M} \right] \end{aligned}$$

2.3. Maximum Likelihood Model

Given the training data, (X_n, y_n), $n = 1, \dots, N$, the GMM parameter set θ can be easily obtained using the maximum likelihood (ML) criterion [11]:

$$\theta_{ML} = argmax_\theta \sum_{n=1}^{N} log p_\theta(X_n|y_n)$$

Using the HTK, we can obtain the ML baseline model parameter set θ_{ML}.

2.4. Discriminative Model

The ML is the most widely-used criterion to obtain the GMM parameter set θ. However, the ML does not have good generalization ability especially when the number of training data is small. In such cases, a discriminative training (DT) criterion shows better result. In this work, we use the following discriminative training criterion [6], 7, 8, 9, 10]:

$$\begin{aligned} \min_{\rho, \boldsymbol{\xi}, \theta} \quad & -\rho + \frac{C}{N} \sum_{n=1}^{N} \xi_n \\ s.t. \quad & F(x_n, y_n; \theta) - F(x_n, y; \theta) \geq \rho - \xi_n \end{aligned}$$

The learning criterion finds the parameter set by maximizing the margin ρ and simultaneously minimizing the sum of slack variables $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_n\}$ so that the difference between the discriminant functions given the correct label and the incorrect labels ($y \in Y \backslash y_n$) is greater than or equal to $\rho - \xi_n$. The parameter C controls the trade-off between the margin maximization and the training error minimization, i.e. sum of slack variables minimization.

2.5. Hierarchical GMM for Task1

In the Task1 dataset, there are 15 scene classes, and some classes have very similar statistics. Thus, we use a 2-level hierarchical label structure where the 15 scene classes are grouped into 4 top classes, and each top class has 3-4 scene class labels. In this structure, we first classify the test input into one of 4 top classes, and then we can finally obtain the scene label by classifying the test input using the 2nd level classifier of the selected top class.

We grouped the 15 scene labels as following table:

Table 1. Four Groups of 15 Scene Classes

Group1	Group2	Group3	Group4
beach forest_path park residential_area	bus car train tram	cafe/restaurant city_center grocery_store metro_station	home library office

In summary, we need 5 classifiers: one classifier for the top level and four classifiers for the 2nd level. In the experiment, the result which does not use the hierarchical structure are also shown for the comparison.

2.6. Binary GMMs for Task4

In contrast to Task1, Task4 is multi-labeled tagging problem, which means each audio clip may have more than one labels among 7 labels in total. To generate multi-labeled tag, we build one vs. the rest binary GMM classifiers for each label. We divide the training samples into positive sets and negative sets. The positive sets consist of the samples that includes the target labels and the negative set consist of the samples that never include the target labels. Then, we train 7 binary classifiers to detect each label independently.

3. EXPERIMENTAL RESULTS

3.1. Task1

The experiments were performed using the Task1 challenge data set with 4 folds split. The MFCC feature vectors were extracted with 20 dimensional cepstrum and the corresponding delta and acceleration coefficients. The frame size and its rate were 40ms and 20ms, respectively. In all experiments, we used 8 mixture components for GMM. The number of mixture component was determined by increasing it until there is no more performance improvement. First, the ML baseline model was obtained, and the θ_{ML} was updated by the DT criterion. Also, the hierarchical label structure was evaluated for each learning criterion. The average classification accuracies of all training methods for four folds were summarized in **Table 2. Classification Accuracy of each Method (%)Table 2**.

Table 2. Classification Accuracy of each Method (%)

	ML	DT	Hierarchical ML	Hierarchical DT
Fold1	72.41	79.31	75.86	84.83
Fold2	66.21	70.34	63.45	77.59
Fold3	72.15	80.54	73.83	86.58
Fold4	79.11	80.48	82.19	86.99
Avg.	72.47	77.67	73.83	84.00

Compared to the ML baseline model, the DT model shows about 5% performance improvement. Comparing the ML model and hierarchical ML model, we observed that there is a small improvement. However, comparing the DT model and hierarchical DT model, we observed that there is a quite much improvement. The hierarchical label structure shows more effective in the DT model than in the ML model.

In the experiments of hierarchical label structure, we obtained the classifier performances which were evaluated separately in each level and evaluated jointly. The performances for ML and DT are shown in **Table 3** and **Table 4**, respectively. In the level 1, the DT model shows 2.7% performance improvement over the ML model. However, in the level 2, the DT model shows much performance improvement (about 9.8%) over the ML model.

Finally, each class accuracy of Hierarchical DT model is shown in **Table 5**. The scene labels of café_restaurant and office show high accuracy (more than 90%) for all folds. However, the scene labels of library and park show low accuracy (less than 40%) in fold 2.

Table 3. The Classifier Performances (%) Evaluated Separately in each Level and Evaluated Jointly for ML

	Level 1	Level 2	Level 1+2
Fold1	92.07	82.07	75.86
Fold2	84.14	76.21	63.45
Fold3	89.60	77.52	73.83
Fold4	96.23	83.90	82.19
Avg.	90.51	79.93	73.83

Table 4. The Classifier Performances (%) Evaluated Separately in each Level and Evaluated Jointly for DT

	Level 1	Level 2	Level 1+2
Fold1	93.10	91.72	84.83
Fold2	90.69	88.85	77.59
Fold3	92.62	89.60	86.58
Fold4	96.58	88.70	86.99
Avg.	93.25	89.72	84.00

Table 5. Class Accuracy (%) of each Fold given Hierarchical DT model

	Fold1	Fold2	Fold3	Fold4
beach	63.16	100.00	94.74	52.63
bus	100.00	50.00	100.00	85.00
cafe_restaurant	100.00	100.00	95.24	90.00
car	95.00	100.00	73.68	100.00
city_center	88.89	73.68	94.74	100.00
forest_path	95.24	100.00	88.89	100.00
grocery_store	78.95	85.71	100.00	84.21
home	95.45	61.11	80.00	88.89
library	71.43	38.89	100.00	100.00
metro_station	73.68	72.22	90.91	100.00
office	94.74	100.00	100.00	100.00
park	100.00	33.33	65.00	60.00
residential_area	84.21	57.14	100.00	89.47
train	38.89	94.74	60.87	50.00
tram	88.89	88.89	63.64	100.00

3.2. Task4

The experiment were performed using the Task4 data with 5 fold split. The MFCC feature vectors were extracted with 20 dimensional cepstrum and corresponding delta and acceleration coefficients. The frame size and its rate were 30ms and 10ms, respectively. We conducted experiments with different numbers of mixtures. **Table 6**, **7**, and **8** shows the results of 16, 32, and 64 mixtures of Gaussian model respectively. The corresponding average EERs were 18.71%, 17.97%, and 17.59%. 64 mixture model performed the best if we choose the same number of mixtures for all labels. However, if we can tune the number of mixtures differently for each label, we could achieve up to 16.71%, which is 0.9% lower than 64 mixture result. **Table 9** shows the performance when the number of mixture is tuned. 64 mixture was the best for adult female speech, other, and video game/tv. 32 mixture was the best for adult male speech, child speech, and percussive sound. 16 mixture was the best for broadband noise. Although the different choice of the number of mixture for each label shows the better performance, we have submitted 64 mixture model result to the challenge, because our goal was to measure the performance of algorithm itself without any manual tuning.

Table 6. Equal Error Rate (%) of 16 mixture model

	ML	DT
adult female speech	27.68	28.46
adult male speech	29.03	23.73
broadband noise	8.18	4.38
child speech	19.16	15.86
other	30.16	31.29
percussive sound	25.14	20.58
video game/tv	6.67	6.67
average	20.86	18.71

Table 7. Equal Error Rate (%) of 32 mixture model

	ML	DT
adult female speech	28.64	26.66
adult male speech	25.79	20.88
broadband noise	7.04	7.10
child speech	16.70	15.19
other	30.57	29.80
percussive sound	25.64	19.92
video game/tv	6.51	6.22
average	20.13	17.97

Table 8. Equal Error Rate (%) of 64 mixture model

	ML	DT
adult female speech	29.14	24.9
adult male speech	25.17	21.61
broadband noise	7.28	7.42
child speech	17.87	16.46
other	30.04	27.85
percussive sound	25.76	21.0
video game/tv	6.56	3.87
average	20.26	17.59

Table 9. Equal Error Rate (%) of best mixture model

	ML (#mixtures)	DT (#mixtures)
adult female speech	27.68 (16)	24.9 (64)
adult male speech	25.17 (64)	20.88 (32)

broadband noise	7.04 (32)	4.38 (16)
child speech	16.70 (32)	15.19 (32)
other	30.04 (64)	27.85 (64)
percussive sound	25.14 (16)	19.92 (32)
video game/tv	6.51 (32)	3.87 (64)
average	19.75	16.71

4. CONCLUSION

In this report, we have proposed a discriminative training algorithm that can be applied to conventional GMM model. After training the baseline GMM model, GMM parameters can be updated by maximizing the margin between classes to improve discriminative characteristics of the model. In the audio classification task, we have further applied hierarchical classifier to discriminate confusing classes. In audio tagging task, we have used multiple binary classifier independently to tag each labels. The proposed model is as simple as GMM, but it has shown significantly improved performance for both tasks.

5. REFERENCES

[1] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993

[2] H. Sak, A. Senior, K. Rao, A. Graves, F. Beaufays, J. Schalkwyk, “Learning acoustic frame labeling for speech recognition with recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015

[3] K.El-Maleh, M. Klein, G. Petrucci, P. Kabal, “Speech/music discrimination for multimedia applications,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2000

[4] E. Benetos and S. Dixon, “Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model,” *J. Acoust. Soc. America*, vol. 133, pp. 1727-1741, 2013

[5] A. Wang, “An industrial strength audio search algorithm,” in *Proc. 4th Int. Conf. Music Inf. Retrieval, Oct.*, 2003, pp. 7-13.

[6] S. Yun and C. D. Yoo, “Loss-scaled large-margin Gaussian mixture models for speech emotion classification,” *IEEE Trans. Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 585-598, 2012.

[7] B. Taskar, C. Guestrin, and D.Koller, “Max-margin Markov networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, vol. 16.

[8] I. Tsochantaridis, T. Joachims, and T. Hofmann, “Large margin methods for structured and interdependent output variables,” *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, 2005.

[9] K. Crammer and Y. Singer, “On the algorithmic implementation of multiclass kernel-based vector machines,” *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2001.

[10] V. Vapnik, *The Nature of Statistical Learning Theory*. New York:Springer, 2000.

[11] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.