Recognition of Sounds Using Square Cauchy Mixture Distribution

Akinori Ito
Graduate School of Engineering
Tohoku University
Sendai, Japan
e-mail: aito@spcom.ecei.tohoku.ac.jp

***Abstract*—In this paper, a new probability density distribution, "the square Cauchy mixture distribution" is proposed for recognition of sound. The proposed density is based on the Cauchy distribution and modified so that it has mean and variance. Since the proposed density can be calculated using only simple arithmetic operations, it can be calculated faster than the Gaussian mixture model (GMM). In addition to the definition of the proposed distribution, a parameter estimation method based on the gradient descent is also described. Two experiments were conducted such as recognition of environmental sound and recognition of singer of the singing voice. The results of the experiments revealed that the proposed method was 10% to 15% faster than the GMM with addlog operation and the recognition performance was comparable.**

Keywords-Gaussian distribution, cauchy distribution, squared cauchy distribution, addlog, environmental sound recognition, singer recognition

I. INTRODUCTION

Statistical modeling of events using Gaussian mixture model (GMM) [1] approximates the probability density of events using weighted sum of the Gaussian distribution. The GMM is widely used for various recognition tasks because of its powerful discrimination ability, theoretical clearness and computational efficiency. In this paper, I consider the target of the recognition task as the recognition of various sounds using a small sensor, such as Internet of Things [2].

The sensors for such purposes are supposed to obtain various information in the environment with small energy consumption and transmit the obtained information through the wireless network. There have been proposed a couple of systems that collects sound-based information in a natural environment [3]–[5]. The bandwidth of wireless network used by such sensors is not so wide, it is desired for a sensor to analyze and compress the obtained sound information. To do this, analysis/compression/recognition methods with small computational load is desired to suppress energy consumption. The GMM is a method that shows high recognition performance with relatively small computation, and thus is widely used for not only recognition of sound [6]–[8] but also recognition of electric appliances using power consumption pattern [9], recognition of video sequence [10] and recognition of user's behavior using a wearable device [11].

There have been several works to achieve fast computation of the GMM, such as quantization of Gaussian distribution and fast log-exp calculation using addlog [12]. Those works are based on the assumption that the distribution is based on the Gaussian distribution. In this paper, I investigated a possibility to use a probability density function that is calculated more easily than the Gaussian distribution. The proposed distribution, "the square Cauchy mixture distribution", can be calculated using only the basic arithmetic operations, yet having at least as powerful discrimination ability as the GMM.

II. PREVIOUS WORK: FAST COMPUTATION OF GMM

A. The Gaussian mixture model

Let an input be a D-dimensional vector $\mathbf{x}=(x_1,\ldots,x_D)$. The probability density of $\mathbf{x}$ by a multivariate Gaussian distribution is as follows.

$$N(\boldsymbol{x};\boldsymbol{\mu},\Sigma)=\frac{\exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}\right)}{(2\pi)^{D/2}\sqrt{|\Sigma|}} \quad (1)$$

Here, $\boldsymbol{\mu}$ and Σ are the mean vector and the covariance matrix, respectively, and $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$. If we assume that the covariance matrix is diagonal, the above distribution can be rewritten as follows,

$$N(\boldsymbol{x};\boldsymbol{\mu},\Sigma)=\prod_{k=1}^{D}\frac{\exp\left(-\frac{(x_k-\mu_k)^2}{2\sigma_k^2}\right)}{\sqrt{2\pi\sigma_k^2}} \quad (2)$$

where μ_k is the k-th component of the mean vector and σ_k^2 is the k-th diagonal component of the covariance matrix.

The Gaussian mixture model is weighted sum of the Gaussian distribution, and it can approximate various probability density by adding many Gaussians. Let M be the number of mixture component. Then the probability density of the GMM can be written as follows.

$$p(\boldsymbol{x})=\sum_{i=1}^{M}\lambda_i N(\boldsymbol{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i) \quad (3)$$

Here, $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ are the i-th mean vector and covariance matrix, respectively, and λ_i is a weight of i-th component where $0 \le \lambda_i \le 1$ and

978-1-5090-2377-6/16/$31.00 ©2016 IEEE

$$\sum_{i=1}^{M} \lambda_i = 1. \tag{4}$$

The parameters of a GMM can be estimated using the EM algorithm [1]. Number of mixture components M is a hyper-parameter to be given. There is also several work to determine M automatically based on Bayes' theory [13].

B. *Fast computation using addlog*

Addlog [12] is a popular technique for faster GMM computation. When calculating Eq. (2), its logarithm is

$$\log N(\boldsymbol{x};\boldsymbol{\mu},\Sigma) = -\sum_{k=1}^{D} \frac{(x_k - \mu_k)^2}{2\sigma_k^2} + C \tag{5}$$

$$C = -\sum_{k=1}^{D} \frac{1}{2} \log (2\pi\sigma_k^2) \tag{6}$$

and thus the log probability density can be calculated using only addition, subtraction, multiplication and division. However, when calculating Eq. (3), we need to convert the probability back to the anti-log domain because the densities are added in the anti-log domain. Addlog operation is used to avoid calculation of exp and log.

Addlog operation is to calculate

$$\text{addlog}(x,y) = \log (\exp x + \exp y). \tag{7}$$

Let we assume $x \geq y$. Then the above operation can be written as follows.

$$\text{addlog}(x,y) = \log (\exp x + \exp y) \tag{8}$$

$$= \log (\exp (x)(1 + \exp (y - x))) \tag{9}$$

$$= x + \log (1 + \exp (y - x)) \tag{10}$$

Here, because $\log (1 + \exp (y - x))$ has bounded value as

$$0 < \log (1 + \exp (y - x)) \leq \log 2 \tag{11}$$

and its value rapidly converges to zero when $|y - x|$ become larger. Thus we can approximate it by making a table of the values of $\log(1+\exp(z))$. With addlog operation, we can calculate the log probability density of the GMM using only simple arithmetic operations and table access.

III. THE SQUARE CAUCHY MIXTURE DISTRIBUTION

A. *From the Cauchy distribution to the square Cauchy distribution*

The Cauchy distribution is a one-dimensional, symmetric and long-tailed distribution with the following probability density.

$$p(x) = \frac{b}{\pi(b^2 + (x - a)^2)} \tag{12}$$

Here, a is the location parameter and b is the scale parameter. Although those parameter determines the location and width of the distribution, the Cauchy distribution does not have any mean and variance because the integrals for calculating those moments do not converge.

Since the Cauchy distribution can be calculated by simple arithmetic operations, calculation of the Cauchy distribution is expected to be fast. However, it is not appropriate to model ordinary event that does not have so much outliers.

Thus, I invented the square Cauchy distribution, which has mean and variance and yet easily calculated using simple arithmetic operations. The square Cauchy distribution is defined as follows.

$$p(x) = \frac{2\sigma^3}{\pi\left(\sigma^2 + (x - \mu)^2\right)^2} \tag{13}$$

The mean and variance of this distribution is μ and σ^2, respectively. The third or higher moments are not exist.

Fig. 1 shows the Gaussian, Cauchy and square Cauchy distributions. We can see that the square Cauchy distribution has higher density at the center and also has longer tail than the Gaussian.

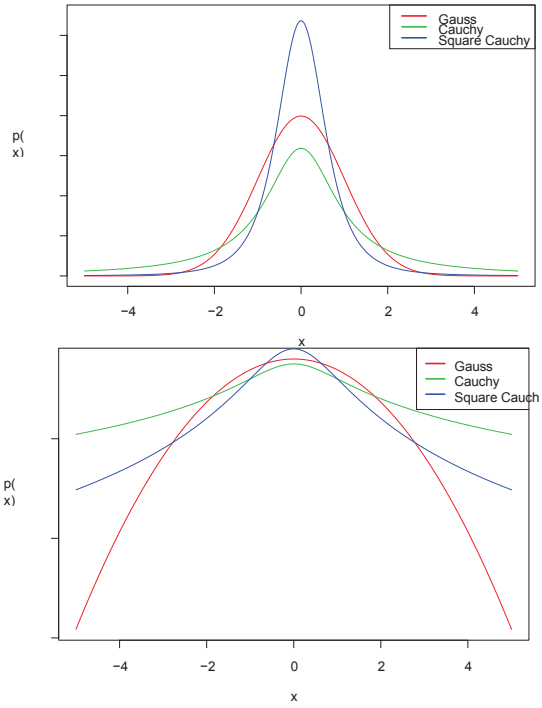

Figure 1. Probability density functions of Gaussian, Cauchy and Square Cauchy distributions (upper: anti-log domain, lower: log domain)

B. *Multivariate Square Cauchy Mixture Distribution*

When the input is a vector, we can compose multivariate squared Cauchy distribution, assuming that there are no correlations between dimensions (as the multivariate Gaussian distribution with diagonal covariance matrix).

$$p(\boldsymbol{x}) = \prod_{k=1}^{D} \frac{2\sigma_k^3}{\pi\left(\sigma_k^2 + (x_k - \mu_k)^2\right)^2} \tag{14}$$

Here,

$$K = \prod_{k=1}^{D} \frac{2\sigma_k^3}{\pi} \tag{15}$$

can be calculated beforehand; thus the calculation becomes

$$p(\boldsymbol{x}) = \frac{K}{\left(\prod_{k=1}^{D} (\sigma_k^2 + (x_k - \mu_k)^2\right)^2} \tag{16}$$

that requires only $2D$ addition/subtraction, $2D$ multiplication and one division. As this calculation is in anti-log domain, we do not need any special operation such as addlog when calculating the mixture density. We need one logarithm operation when calculating log probability.

$$p(\boldsymbol{x}) = \sum_{i=1}^{M} \frac{K_i}{\left(\prod_{k=1}^{D} (\sigma_{ik}^2 + (x_k - \mu_{ik})^2\right)^2} \tag{17}$$

$$K_i = \lambda_i \prod_{k=1}^{D} \frac{2\sigma_{ik}^3}{\pi} \tag{18}$$

The order of arithmetic operation of the proposed distribution and the GMM with addlog operation is comparable, but the proposed distribution has an advantage that it does not need to have a table for calculation.

C. Parameter Estimation of the Square Cauchy Distribution

Although the square Cauchy distribution has mean and variance, the sample mean and variance are not necessarily the best estimation from the maximum likelihood point of view. Thus, a maximum likelihood parameter estimation method is derived. The estimation is basically a gradient descent method.

We rewrite Eq. (17) as follows.

$$p(\boldsymbol{x}) = \sum_{i=1}^{M} \lambda_i p_i(\boldsymbol{x}) \tag{19}$$

$$p_i(\boldsymbol{x}) = \prod_{k=1}^{D} \frac{2\sigma_{ik}^3}{\pi\left(\sigma_{ik}^2 + (x_k - \mu_{ik})^2\right)^2} \tag{20}$$

By differentiating Eq. (19) by μ_{ik}, σ_{ik}^2 and λ_i, we obtain

$$\frac{\partial p(\boldsymbol{x})}{\partial \mu_{ik}} = \frac{4\lambda_i (x_i - \mu_{ik})}{(x_i - \mu_{ik})^2 + \sigma_{ik}^2} p_i(\boldsymbol{x}) \tag{21}$$

$$\frac{\partial p(\boldsymbol{x})}{\partial \sigma_{ik}^2} = \frac{\lambda_i \ \ 3(x_i - \mu_{ik}) - \sigma_{ik}^2}{2\sigma_{ik}^2 (x_i - \mu_{ik})^2 + \sigma_{ik}^2} p_i(\boldsymbol{x}) \tag{22}$$

$$\frac{\partial p(\boldsymbol{x})}{\partial \lambda_i} = p_i(\boldsymbol{x}) \tag{23}$$

Thus,

$$\frac{\partial \log p(\boldsymbol{x})}{\partial \mu_{ik}} = \frac{1}{p(\boldsymbol{x})} \frac{\partial p(\boldsymbol{x})}{\partial \mu_{ik}} \tag{24}$$

$$\frac{\partial \log p(\boldsymbol{x})}{\partial \sigma_{ik}^2} = \frac{1}{p(\boldsymbol{x})} \frac{\partial p(\boldsymbol{x})}{\partial \sigma_{ik}^2} \tag{25}$$

$$\frac{\partial \log p(\boldsymbol{x})}{\partial \mu_{ik}} = \frac{p_i(\boldsymbol{x})}{p(\boldsymbol{x})} \tag{26}$$

When we have the log likelihood $L(X)$ for the training data $X = \mathbf{x}_1, \ldots, \mathbf{x}_N$ as

$$L(X) = \sum_{n=1}^{N} \log p(\boldsymbol{x}_n), \tag{27}$$

then

$$\frac{\partial L(X)}{\partial \mu_{ik}} = \sum_{n=1}^{N} \frac{\partial \log p(\boldsymbol{x}_n)}{\partial \mu_{ik}} \tag{28}$$

and we can update the parameter μ_{ik} as

$$\mu_{ik}' \leftarrow \mu_{ik} + \eta \frac{\partial L(X)}{\partial \mu_{ik}} \tag{29}$$

where $\eta > 0$ is the learning rate. σ_{ij}^2 and λ_i can also be updated in the same way. When estimating the parameters, mean, variance and weight of the GMM are good initial values. Thus, we first train a GMM for given training data, and then update the parameters using the above-mentioned method. In the following experiments, the learning rate η was determined by Adam [14]. To avoid the variance to become smaller or negative, update was stopped when the variance became smaller than 0.03.

IV. Experiment

A. Recognition of Environmental Sounds

Two experiments were conducted to compare the GMM and the proposed method. The first experiment is recognition of environmental sounds. The task was to discriminate 14 environmental sounds taken from JEITA Noise Database. The sounds were recorded in real environments such as station, factory, roadside etc. 300 seconds of signal of each environment were used as training data, and 20 signals of 10 seconds for each of the environment were used as test data. All the signals were recorded as 16 kHz, 16 bit quantization. The signals were converted into 13-dimensional MFCC (without Δ) using 40 ms Hamming window and 10 ms frame shift. Two methods were compared for the proposed method; the first one was to use the means and variances calculated for GMM, and the second one was calculated through the optimization shown as Eq. (29).

In addition to the GMM and the square Cauchy mixture, a simple vector quantization (VQ)-based method is also tested. In this method, we first calculated clusters for each of the training data using the k-means algorithm, and the given number of cluster centroids were calculated. When an evaluation data was given, we choose the nearest centroid to a certain frame and determined the class of the frame according to the class the nearest centroid belonged to. Finally, class of the evaluation data was determined based on majority vote for decision results of all frames. This method

only requires the distance calculation to the all centroids, and thus is expected to be faster than the GMM and the proposed method, but is also expected to be less accurate compared to other methods when the number of centroids are small.

Figure 2 shows the accuracy for various number of mixture components (or number of centroids for VQ). Here, label "SqCauchy(GMM)" is the squared Cauchy mixture model using means and variances for GMM, and "SqCauchy(SCMM)" is that using the optimized means and variances. We can see that GMM and the proposed method could classify the evaluation data almost accurately using only one distribution, whereas the VQ-based method needs two centroids for each class for accurate classification, as expected. Because the recognition accuracy was high enough, no significant difference was observed between the square Cauchy mixture distribution using the GMM's parameters and that using the optimized ones.

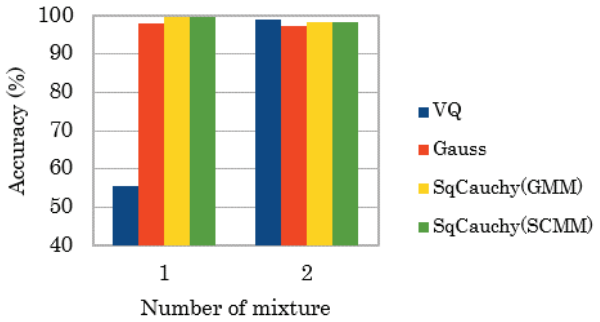

Figure 2. Recognition result of environmental sound

Next, the computation times of the tested methods were compared. The time for computing log probability (or distances) of all frames of the evaluation data (14 sounds × 10 seconds × 20 clips = 2400 seconds). Time for input/output and feature extraction is not included. Raspberry Pi Model B+ (ARM1176 core, 700 MHz, 512MByte memory) with Raspbian OS was used for computation. The recognition experiment was repeated 10 times and the average time was computed.

Figure 3 shows the results. Four methods were compared: GMM without addlog (label Gauss), GMM with addlog (label Addlog) and the proposed method (label Square Cauchy). For single distribution case, addlog and the proposed method showed almost the same speed. When the number of mixture was 2, the proposed method was 1.3 times faster than the addlog-based GMM.

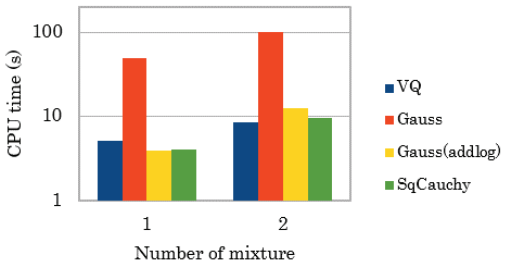

Figure 3. Comparison of CPU time (experiment 1)

B. *Recognition of Singer of the Singing Voice*

Next, a little more difficult task is examined. This task is to discriminate 16 amateur singers from two seconds of singing voice. The corpus for the experiment is corpus of singing voice of Japanese karaoke [15]. To obtain high accuracy, the state-of-the-art method such as i-vector [16] should be used; however, the purpose of this experiment is to compare the performance of the proposed model and the GMM, a simple framework [17] is employed in this experiment.

First, a universal background model (UBM) is trained as a GMM from the database using 11 male singers in which each singer sang two individual songs. Four phrases were extracted from one song, each of which was 4.45 s in average, and the phrases were used for the training data of the UBM. Next, singers' models were trained using the UBM as an initial model and singing voice of 16 male singers as the adaptation data. Each singer sang the same song (*Itoshi no Ellie*, or Ellie my love) twice, and 47 s of singing voice from that song was used as the adaptation data of that singer. Evaluation data were taken from the same singing voice as the adaptation data, but there were no overlap between the adaptation and evaluation data. The amount of evaluation data for one singer was about 2 s.

All the signals were recorded as 16 kHz, 16 bit quantization. The signals were converted into 13-dimensional MFCC with Δ and ΔΔ (39 dimensions in total) using 40 ms Hamming window and 10 ms frame shift.

Figure 4 shows the result. Different from the previous result, the proposed method (label Square Cauchy) shows lower recognition accuracy for smaller number of mixture than the GMM (label Gauss), and the accuracy for the larger number of mixture was the same. The recognition accuracy of the proposed method raised, and the accuracy of the method at 64 mixture was as good as that of the GMM. The optimized parameter for the square Cauchy mixture distribution was effective for all mixture. VQ-based method showed high accuracy at 16 centroids, but it did not give the best accuracy.

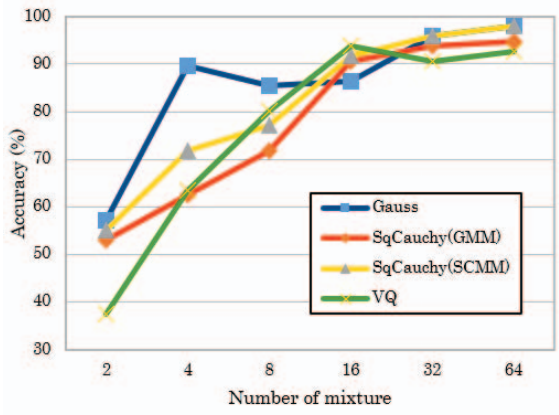

Figure 4. Recognition result of singers

Next, computation time for each number of mixture was measured. The time for computing log probability of all

frames of the evaluation data (16 singers × 2.1 seconds = 33.6 seconds). Time for input/output and feature extraction is not included. The computer used in this experiment was the same as that of the previous experiment.

Fig. 5 shows the results. Three methods were compared: GMM with addlog (label Gauss (addlog)), the proposed method (label Square Cauchy) and the VQ-based method (label VQ). The proposed method was 10% to 15% faster than the addlog-based GMM. VQ was almost twice as fast as the proposed method.

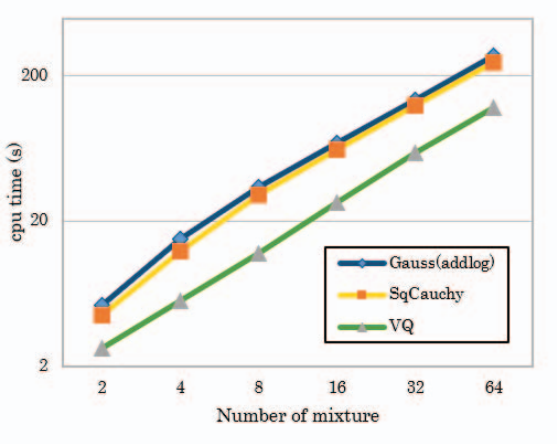

Figure 5. Comparison of CPU time (experiment 2)

V. SUMMARY

In this paper, a new probability distribution "square Cauchy mixture distribution" was proposed for the use of probabilistic model for recognition of sound. The proposed density is easier to compute, yet having comparable recognition performance to the GMM. From the two recognition experiments, it became clear that the proposed method was 10% to 15% faster than the GMM and the recognition accuracy was at least as good as that of the GMM.

REFERENCES

[1] R. A. Redner and H. F. Walker, "Mixture densities, maximum likelihood and the EM algorithm," SIAM Review, vol. 26, no. 2, pp. 195–239, 1984.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future Generation Computer Systems, vol. 29, no. 7, pp. 1645 – 1660, 2013.

[3] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and yield in a volcano monitoring sensor network," in Proc. the 7th Symposium on Operating Systems Design and Implementation, ser. OSDI '06, 2006, pp. 381–396.

[4] N.-H. Liu, C.-A. Wu, and S.-J. Hsieh, "Long-term animal observation by wireless sensor networks with sound recognition," in Wireless Algorithms, Systems, and Applications, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5682, pp. 1–11.

[5] R. Gonzalez and Y. Gao, "An audiovisual wireless field guide," in The Era of Interactive Media. Springer New York, 2013, pp. 631–641.

[6] M. Cowling and R. Sitte, "Comparison of techniques for environmental sound recognition," Pattern Recognition Letters, vol. 24, no. 15, pp. 2895 – 2907, 2003.

[7] S. Chu, S. Narayanan, and C.-C. Kuo, "Environmental sound recognition with time-frequency audio features," IEEE Transactions on Audio, Speech, and Language Processing,, vol. 17, no. 6, pp. 1142–1158, Aug 2009.

[8] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Automatic recognition of urban soundscenes," in New Directions in Intelligent Interactive Multimedia. Springer Berlin Heidelberg, 2008, vol. 142, pp. 147–153.

[9] Y.-X. Lai, C.-F. Lai, Y.-M. Huang, and H.-C. Chao, "Multi-appliance recognition system with hybrid SVM/GMM classifier in ubiquitous smart home," Information Sciences, vol. 230, pp. 39–55, 2013.

[10] S.-H. Ou, C.-H. Lee, V. S. Somayazulu, Y.-K. Chen, and S.-Y. Chien, "Video sensor node with dis1t ributed video summary for internetof-things applications," in Proc. Int. Conf. on Consumer ElectronicsTaiwan, 2015.

[11] P. Meharia and D. P. Agrawal, "The able amble: gait recognition using Gaussian mixture model for biometric applications," in Proc. the 12th ACM Int. Conf. on Computing Frontiers, 2015.

[12] S. Sagayama and S. Takahashi, "On the use of scalar quantization for fast HMM computation," in ICASSP-95, vol. 1, May 1995, pp. 213–216 vol.1.

[13] S. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian approaches to Gaussian mixture modeling," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 20, no. 11, pp. 1133–1142, Nov 1998.

[14] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. the 3rd Int. Conf. for Learning Representations, 2015.

[15] R. Daido, M. Ito, S. Makino, and A. Ito, "Automatic evaluation of singing enthusiasm for karaoke," Computer Speech & Language, vol. 28, no. 2, pp. 501–517, 2014.

[16] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Transactions on Audio, Speech and Language Processing, vol. 19, no. 4, pp. 788–798, May 2010.

[17] D. Reynolds and R. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," IEEE Trans. on Speech and Audio Processing, vol. 3, no. 1, pp. 72–83, Jan 1995.