GIF-SP: GA-based Informative Feature for Noisy Speech Recognition

Satoshi Tamura*, Yoji Tagami* and Satoru Hayamizu*
* Department of Information Science, Gifu University, Gifu, Japan
E-mail: tamura@info.gifu-u.ac.jp, tagami@asr.info.gifu-u.ac.jp, hayamizu@gifu-u.ac.jp

***Abstract*—This paper proposes a novel discriminative feature extraction method. The method consists of two stages; in the first stage, a classifier is built for each class, which categorizes an input vector into a certain class or not. From all the parameters of the classifiers, a first transformation can be formed. In the second stage, another transformation that generates a feature vector is subsequently obtained to reduce the dimension and enhance recognition ability. These transformations are computed applying genetic algorithm. In order to evaluate the performance of the proposed feature, speech recognition experiments were conducted. Results in clean training condition shows that GIF greatly improves recognition accuracy compared to conventional MFCC in noisy environments. Multi-condition results also clarifies that out proposed scheme is robust against differences of conditions.**

I. Introduction

For pattern recognition, it is essential and important to extract feature vectors before recognition; the precision, accuracy or performance of a recognition method strongly relies on a feature extraction method and its resulting features. As an example of features in pattern recognition, in speech recognition Mel Frequency Cepstral Coefficients (MFCCs) are widely and commonly used as an acoustic feature. Another instance is easily found in a computer vision domain; in order to extract a particular object in an image or to detect whether an image contains a certain object, various features have been proposed and actually used: e.g. Histogram of Gradients (HoG) [1] and Scale Invariant Feature Transform (SIFT) [2]. These conventional features are based on signal processing techniques. On the other hand, discriminative or adaptive feature extraction methods have attracted attention; one of the typical discriminative feature is obtained by applying Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and Non-Linear Discriminant Analysis (NLDA) [3], which has been employed in many recognition tasks. Or, for noise-robust speech recognition, feature-space Minimum Phone Error (fMPE) [4] and feature-space Maximum Mutual Information (fMMI) [5] have been proposed. These discriminative features are inspired by model adaptation techniques, in which an input feature is projected into another feature space so as to improve recognition accuracy.

This paper proposes a novel feature extraction method based on discriminative techniques. Our proposed method consists of two transformations; the first transformation is constructed to distinguish every classes effectively, while the second is generated to reduce the dimension of feature space, make an entire projection linear independent, and enhance the robustness against noises. In order to obtain these transformations without any restrictions, we employ Genetic Algorithm (GA). In speech processing, there are several works in which GA is used so as to improve acoustic features [6], [7], [8]. Compared to those methods, our proposed method can be easily applied to most Automatic Speech Recognition (ASR) systems with achieving significant improvements. In this paper, our proposed feature is called GIF (GA-based Informative Feature). Our goal is to develop the best features optimized for each pattern recognition task only using a class label and without any prior knowledge about the task, employing possible feature extraction functions and their combination: e.g. linear, non-linear (high-degree polynomial function, logarithm and exponential functions, etc), differentiable, non-differentiable, and if-then functions. This work is therefore out first attempt to the goal.

This paper is organized as follows: Section II introduces our proposed feature extraction in detail. We test our proposed feature GIF using an evaluation corpus, then experimental setup and results as well as discussion are appeared in Section III. And finally Section IV concludes this paper.

II. GA-based Informative Feature

In this section, our proposed feature GIF is introduced. This feature can be utilized in various pattern recognition tasks and related works [9], [10].

Let us denote an N-dimensional input feature space by $\mathbb{X}$ and a feature vector (e.g. filter-bank amplitudes) by $\boldsymbol{x} \in \mathbb{X}$. Now a transformation is considered to project an input vector into a new feature space for recognition, and in this paper, a linear transformation is assumed.

At first, an input vector $\boldsymbol{x}$ is converted into a C-dimensional intermediate vector $\boldsymbol{y}$ as:

$$\boldsymbol{y} = A \left(\boldsymbol{x}^\top \ 1\right)^\top \tag{1}$$

In Eq.(1), A is a $C \times (N+1)$ transformation matrix, where C is the number of classes that should be classified (e.g. phonemes). This process is called Stage 1. Next, the vector $\boldsymbol{y}$ is converted into an output feature vector (GIF) that will be used in a speech recognizer. If we denote an M-dimensional feature space for recognition by $\mathbb{Z}$ $(1 \leq M \leq C)$, a feature vector $\boldsymbol{z} \in \mathbb{Z}$ is obtained as:

$$\boldsymbol{z} = B \ \boldsymbol{y} \tag{2}$$

where B is an $M \times C$ transformation matrix (Stage 2). By separating feature transformation into two stages (for class discrimination and dimension reduction / orthogonalization), the matrices can be easily obtained.

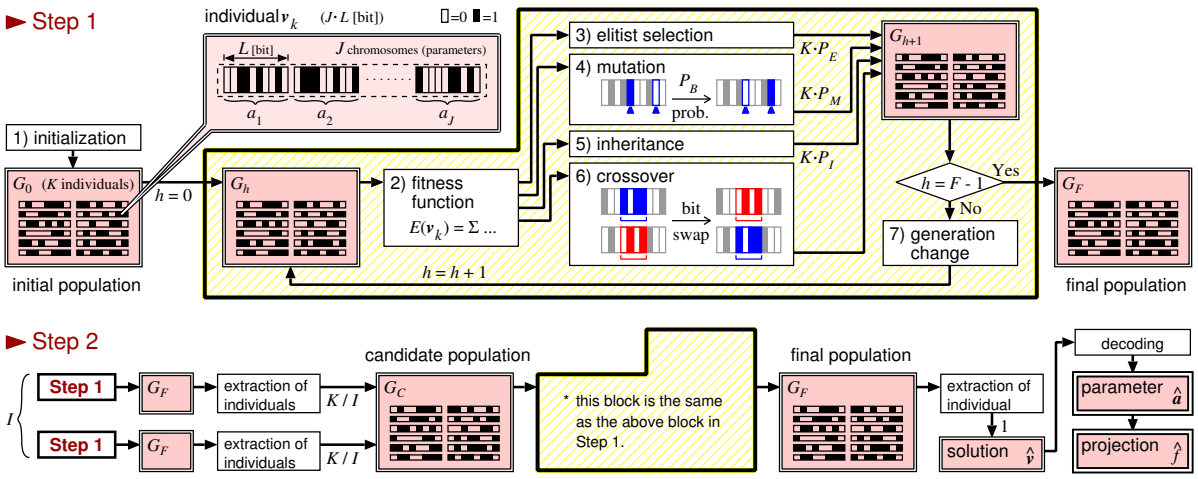

Fig. 1. Computation of projection parameters of a binary classifier in Stage 1 (see Section II-A).

A. *Stage 1: Getting a first transformation matrix*

A binary classifier which distinguishes an input vector into a certain class or its complementary class (e.g. a class for a vowel and a class for the other phonemes) is focused on. Figure 1 illustrates a flow of computing the binary classifier and obtaining a part of the matrix $A = (a_{i,j})$. The transformation matrix A is eventually obtained by applying this processing for all classes.

1) Step 1: Building a candidate classifier.:

For the i-th class $(1 \leq i \leq C)$, assume a linear classifier f for an input vector $\boldsymbol{x} = (x_j)$:

$$f(\boldsymbol{x}; \boldsymbol{a}_i) = \left(\sum_{j=1}^{N} a_{i,j} x_j \right) + a_{i,N+1} \tag{3}$$

where $\boldsymbol{a}_i = (a_{i,1}, \cdots, a_{i,N}, a_{i,N+1})$ includes classifier parameters corresponding to a part of A. The classifier is designed to return a positive value if $\boldsymbol{x}$ is in the class or a negative value otherwise. In our proposed method, the classifier parameters are obtained using a training set $R = \{\boldsymbol{r}_n\} \subset \mathbb{X}$ and the standard genetic algorithm.

1. Initialization
   Create an initial population G_0 including K individuals. An individual has J $(=N+1)$ chromosomes, each of which means an L-bit floating-point value encoding a classifier parameter.
2. Fitness function
   For the k-th individual $\boldsymbol{v}_k$ in the h-th generation G_h, a fitness function $E(\boldsymbol{v}_k; \boldsymbol{a})$ is calculated as:
   $$E(\boldsymbol{v}_k; \boldsymbol{a}) = \sum_{n=1}^{|R|} l_n \cdot sgn\big(f_i(\boldsymbol{r}_n; \boldsymbol{a})\big) \tag{4}$$
   where $\boldsymbol{a}$ is a parameter set obtained by decoding $\boldsymbol{v}_k$, and l_n is a transcribed label that corresponds to 1 if $\boldsymbol{r}_n$ belongs to the class or -1 otherwise. If the value $E(\boldsymbol{v}_k; \boldsymbol{a})$ becomes zero or negative, the fitness function value is set to 1.
3. Elitist selection
   From a current generation G_h, choose individuals that have the $(K \cdot P_E)$ highest fitness function values and add them to a next generation G_{h+1}.
4. Mutation
   In the following operations, a probability of which an individual $\boldsymbol{v}_k$ is selected is formulated as:
   $$P(\boldsymbol{v}_k; \boldsymbol{a}) = E(\boldsymbol{v}_k; \boldsymbol{a}) / \textstyle\sum_{m=1}^{K} E(\boldsymbol{v}_m; \boldsymbol{a}) \tag{5}$$
   Choose an individual from G_h, conduct a mutation operation, and add an obtained individual to G_{h+1}; this operation inverts several bits in the individual, where the inversion probability is P_B. This operation is repeated $(K \cdot P_M)$ times.
5. Inheritance
   Choose an individual from G_h and directly add it to G_{h+1}. This operation is repeated $(K \cdot P_I)$ times.
6. Crossover
   Choose two individuals from G_h, and a crossover operation is conducted to obtain new individuals which should be added to G_{h+1}; a chromosome index in the individual and bit indexes d_1 and d_2 are randomly chosen $(1 \leq d_1 < d_2 \leq L)$, then bit swapping between d_1-th and d_2-th bits in the chromosome is performed. This operation is repeated until the number of individuals in G_{h+1} reaches K.
7. Generation change
   The above processing (from 2. to 6.) is repeatedly conducted from $h = 0$ to $h = F - 1$. Consequently a final population G_F is obtained.

2) Step 2: Completing a binary classifier:

After Step 1 is performed, individuals that have the (K/I) highest fitness function values in G_F are extracted and added to a candidate population G_C By repeating the process I times, the candidate population is completed. Step 1 is again applied where G_C is used as the initialized generation, then the best-fit individual $\hat{\boldsymbol{v}}$ is obtained. The transformation parameter set $\hat{\boldsymbol{a}}$ can be finally acquired by decoding the solution $\hat{\boldsymbol{v}}$. It

is often pointed out that solution by GA is not stable since the operations in GA highly depend on an initial population randomly generated. By employing the two-step approach, relatively stable solution can be obtained.

B. Stage 2: Getting a second transformation matrix

To enhance the discriminative and recognition ability, and to reduce the dimension of feature vectors, we employ a second-stage procedure explained below:

1. For the i-th class, calculate a C-dimensional mean vector $\bar{\boldsymbol{\mu}}_i$:
$$\bar{\boldsymbol{\mu}}_i = \frac{1}{|R_i|} \sum_{\boldsymbol{x} \in R_i} A(\boldsymbol{x}^\top \ 1)^\top \tag{6}$$
where R_i is a subset of the training data, where all vectors belong to the i-th class.
2. Let us denote a linear transformation g for an vector $\boldsymbol{y} = (y_j)$ obtained in the first stage by the following equation:
$$g(\boldsymbol{y}; \boldsymbol{b}_m) = \sum_{j=1}^{C} b_{m,j} y_j \tag{7}$$
where $\boldsymbol{b}_m = (b_{m,1}, \cdots, b_{m,C})$ indicates classifier parameters and a part of B.
3. For $m=1$ in Eq.(7), the linear projection g is determined so that a variance of transformed values would be maximized. The parameter set $\boldsymbol{b}_1$ is then optimized by applying the GA explained previously, where the fitness function is modified as:
$$E(\boldsymbol{v}_k; \boldsymbol{b}) = var(w_1, \cdots, w_C) \quad , \quad w_i = g(\bar{\boldsymbol{\mu}}_i; \boldsymbol{b}) \tag{8}$$
In Eq.(8), $\boldsymbol{b}$ is obtained by decoding $\boldsymbol{v}_k$.
4. For $m = 2$, $\boldsymbol{b}_2$ can be obtained so as to maximize a variance just as same as $\boldsymbol{b}_1$, under the constraint that an inner product between $\boldsymbol{b}_1{}^\top$ and $\boldsymbol{b}_2{}^\top$ is zero:
$$\left\langle \boldsymbol{b}_1{}^\top, \boldsymbol{b}_2{}^\top \right\rangle = \sum_{j=1}^{C} b_{1,j} b_{2,j} = 0 \tag{9}$$
5. For any m $(2 \leq m \leq M)$, the m-th parameter set $\boldsymbol{b}_m$ can be calculated in the same way, under the restriction that all inner products should be zero.

C. Feature vector computation

Once the first projection A and the second projection B are determined, a feature vector $\boldsymbol{z}$ can be obtained by applying Eqs.(1) and (2). Before applying the second projection, the bias vector $\boldsymbol{\mu}$ is computed beforehand as:
$$\boldsymbol{\mu} = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{y}_t = \frac{1}{T} \sum_{t=1}^{T} A \boldsymbol{x}_t \tag{10}$$
where $X = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T)$ is a sequence of extended input vectors. Subsequently, each intermediate vector is normalized by suppressing the bias vector:
$$\hat{\boldsymbol{y}}_t = \boldsymbol{y}_t - \boldsymbol{\mu} \tag{11}$$
This process is expected to increase the robustness of the proposed feature, similar to Cepstrum Mean Subtraction (CMS).

TABLE I
EXPERIMENTAL CONDITIONS.

Corpus	CENSREC-1 (Japanese connected digits)
Analysis settings	frame-length$=25$[ms], frame-shift$=10$[ms], hamming window, emphasis-coef$=0.97$, filterbank-channel$=23$ $(N=23)$
Model training	subject$=110$, utterance$=8440$, clean training, HMM-state$=16$(digit HMM), 3(silence HMM)
Testing	subject$=104$, utterance$=4004$
Noise settings	kind=airport, babble, car, exhibition, restaurant, station, street(G.712, MIRS), subway(G.712, MIRS) SNR$=-5$, 0, 5, 10, 15, 20dB, clean
GIF training	class$=18$ $(C=18)$, frame$=13600$ (per class)
GA param.	$L=32$, $K=1000$, $F=30$, $I=50$, $P_B=0.125$, $P_E=0.05$, $P_M=0.01$, $P_I=0.2$

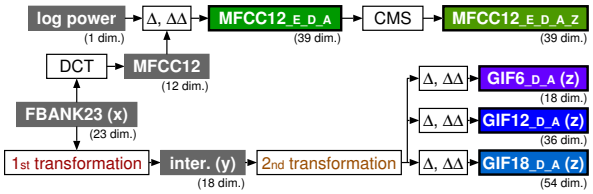

Fig. 2. Conventional and proposed features.

III. EXPERIMENT

In order to evaluate our proposed features, speech recognition experiments were conducted using an evaluation corpus CENSREC-1 (AURORA-2J) [11].

A. Experimental setup

Table I summarizes experimental conditions. As an input feature, 23-dimensional filter-bank coefficients were commonly used ($N = 23$). A 16-state Hidden Markov Model (HMM) was built for each digit, while a 3-state HMM was employed for silence. We conducted two kinds of experiments: clean condition and multi condition. In the clean condition, all HMMs and transformation matrices were built using clean training data. On the other hand, not only clean but also noisy speech data were employed as training data in the multi condition. In Stage 1, 17 phonemes appeared in Japanese digits as well as silence were used as the classes $(C=18)$.

B. Experiment in clean training

In order to evaluate effectiveness of our proposed feature GIF, recognition experiment using clean-training model were conducted. In Stage 1, the number of frames in each phoneme class and in its complementary class was 13,600 (called T_c data set). The same data set was used for Stage 2. These frames were randomly extracted from the training data according to a forced-alignment result. Several conventional features (**MFCC12_E_D_A** and **MFCC12_E_D_A_Z**) as well as proposed features (**GIF6_D_A**, **GIF12_D_A** and **GIF18_D_A**) were tested (see Figure 2). The performance of each feature was evaluated by an average accuracy in 10 testing (noisy) conditions, and the accuracy was obtained at seven SNR levels (-5, 0, 5, 10, 15, 20dB and clean) respectively.

Figure 3 depicts recognition accuracies for every features. Our proposed features drastically improves the accuracies in all noisy conditions: roughly 60% relative error reduction at 20dB and 15dB, or, approximately 20% absolute improvement

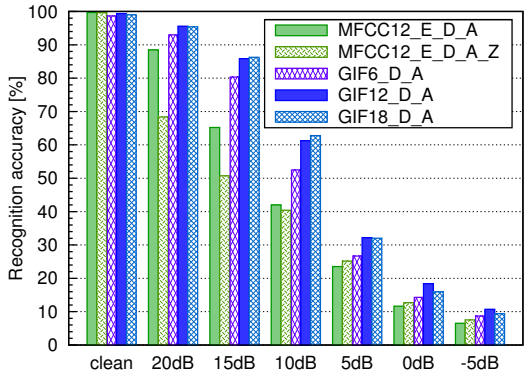

Fig. 3. Recognition accuracies of each feature in the clean training condition.

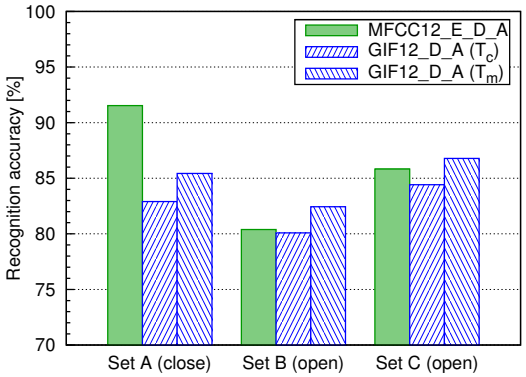

Fig. 4. Average recognition accuracies of each feature for every test data sets in the multi-condition training.

of recognition accuracy at 15dB and 10dB, compared with **MFCC12_E_D_A**. Note that **MFCC12_E_D_A_Z** cannot improve the accuracy from **MFCC12_E_D_A** due to insertion errors in silence periods (before or after utterances).

According to the results, there is few difference between **GIF12_D_A** and **GIF18_D_A**. However, it means that dimension reduction was successfully conducted keeping the recognition performance. In addition, orthogonalization is essential for the conventional HMM scheme. These indicate the effectiveness of the second transformation.

Previous methods using GA, mentioned in Section I, fundamentally try to improve feature extraction based on HMM likelihoods or LDA results. It means these methods cannot be applied to other tasks in which such the architectures are not employed. In contrast, our method can be generally used. This becomes an advantage of our scheme. In order to further indicate the impact of our method, evaluation of competitive features such as LDA with MFCC or fMPE is needed. It is included in our future works.

Finally, regarding computational cost, a lot of computational power is needed in the training stages: roughly dozens of hours even using a parallel-computing architecture. On the other hand, when applying the transformation to obtain a feature, real-time computation is easily accomplished just as MFCC, because only simple linear calculations are required.

C. Experiment in multi-condition training

Since CENSREC-1 provides muti-condition training data, the second experiment was conducted using the data. Similar to model construction, multi-condition training data for GIF transformation were also collected (called T_m data set); for each phoneme 6,800 frames were extracted from the data set used in the previous experiment, in addition to 6,800 frames of speech data overlapped with white noise. **MFCC12_E_D_A** and **GIF12_D_A** were tested in this experiment.

Figure 4 shows recognition results for three test sets in CENSREC-1: set A for closed condition, set B and C for open condition. The performance of **GIF12_D_A** was not sufficient compared to **MFCC12_E_D_A** in Set A, on the other hand, **GIF12_D_A** outperformed **MFCC12_E_D_A** in open conditions. In general, MFCC can achieve high recognition accuracy in the training environments, however, the accuracy degrades in the different conditions. It can be concluded according to the experiments, that the proposed feature GIF has the robustness against condition change including background noise and channel distortions. This becomes an advantage of GIF. It is also found that the performance using the T_m data set was superior to T_c. Using noisy speech data in the GIF training thus makes more robust transformation.

IV. Conclusion

This paper proposes a discriminative feature (GIF) extraction method in which feature transformations are optimized by genetic algorithm. Speech rcognition experiment in clean condition shows that the effectiveness of our method, achieving 20% improvement of recognition accuracy or 60% relative error reduction in middle SNR environments. Through multi-condition experiment, it turns out that GIF is a robust feature against noise or channel differences.

Our future work includes (1) comparison to other relative methods, (2) further investigation of transformation (e.g. non-linear projection), and (3) evaluation of the proposed method in real noisy environments and Large Vocabulary Continuous Speech Recognition (LVCSR) tasks.

References

[1] N.Dalal et al., "Histograms of oriented gradients for human detection," Proc. CVPR2005, vol. 1, pp.886-893 (2005).

[2] D.G.Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, vol.60, no.2, pp.91-110 (2004).

[3] H.Hu et al., "Dimensionality reduction methods for HMM phonetic recognition," Proc.ICASSP2010, pp.4854-4857 (2010).

[4] D.Povey et al., "fMPE: discriminatively trained features for speech recognition," Proc. ICASSP2005, pp. 961-964 (2005).

[5] D.Povey et al., Boosted MMI for model and feature-space discriminative training," Proc. ICASSP2008, pp.4057-4060 (2008).

[6] Z.Behzad et al., "Discriminative transformation for speech features based on genetic algorithm and HMM likelihoods," IEICE Electronics Express, vol.7, no.4, pp.247-253 (2010).

[7] C.Charbuillet et al., "A new approach for speech feature extraction based on genetic algorithms," Proc. ICASSP2007, pp.285-288 (2007).

[8] H.Abbasian et al., "Class dependent LDA optimization using genetic algorithm for robust MFCC extraction," Proc. INTERSPEECH2008, pp.1541-1544 (2008).

[9] N.Ukai et al., 'GIF-LR: GA-based informative feature for lipreading," Proc. APSIPA ASC 2012 (2012).

[10] K.Sawada et al., 'Statistical voice conversion using GA-based informative feature," Proc. APSIPA ASC 2012 (2012).

[11] S.Nakamura et al., "AURORA-2J: an evaluation framework for Japanese noisy speech recognition," IEICE Trans. on Information and Systems, vol.E88-D, no.3, pp.535-544 (2005).