Identification of Birds' Kinds from Songs Using DP Matching and SOM

Kengo Yoshida† and Masahiro Tanaka‡

†Graduate School of Department of Information Science and Systems Engineering
Konan University

8-9-1 Okamoto, Higashinada-ku, Kobe 658-8501, Japan.

E-mail: mn224007@center.konan-u.ac.jp

‡Department of Information Science and Systems Engineering, Konan University

8-9-1 Okamoto, Higashinada-ku, Kobe 658-8501, Japan

E-mail: m.tanaka@konan-u.ac.jp

Abstract

In this paper, the identification of birds' kinds from their acoustic sound is treated. The short time Fourier Transform will be used as the feature vectors. The standard songs will be registered as the template songs. For the recognition problem, DP matching will be tried, and it will be shown to be useful if the songs are of full length as the templates. Next, for the worse condition of the songs, SOM will be used for the sound abstraction and ordering.

1 Introduction

Recently, "bird watching" has become a very popular hobby for the people living in urban area. It is said to be very good for healing stress diseases. If the watchers could identify the birds' kinds easily from the songs, they may have additive interest in it. However, it is difficult for them to identify the birds' kind by merely hearing the birds' songs without looking at them if they are not familiar with the birds. In this research, the identification of birds' kinds is treated.

2 Data

2.1 Data Source

In our experiment, the samples of the songs of known birds are taken from music CDs [6], but it may be also possible to use the songs recorded from the microphones in parks or woods. The songs in CDs are usually good in quality. So, we take them as the template songs that are to be used to identify the birds' kinds.

2.2 Feature Vector

The songs of the same kinds of birds are heard to be similar. However, they are different in the number of the phrases and the total length. By considering the nature of the problem, this can be treated as the pattern-matching problem of similar series of feature vectors.

It is natural to use the spectral information as the feature vector. The spectral information is extracted as follows. The sampling time is 44,100Hz, the window size for short-term FFT is $N (= 512)$ points. Next window begins at $N/2$ points ahead.

To get the spectrum envelope, we use the liftering method [7]. The algorithm is as follows.

1. Take N sample points from the data.
2. Apply Hamming window.
3. Apply liftering (FFT $\rightarrow$ take log $\rightarrow$ IFFT $\rightarrow$ Use $M (= 64)$ points from the top).
4. FFT(M points) $\rightarrow$ Take exponential $\rightarrow$ Use power spectrum at $n (= 20)$ frequencies as the feature vector.

Thus, a song of a bird is transformed into a series of $n (= 20)$ -dimensional feature vector $x(k)$ for every 5.8ms approximately. k is the discrete-time. That is, a song S can be expressed as

$$S = [x(1), x(2), \dots, x(N_S)]$$

where $x(k) \in \mathbb{R}^n$ ($n = 20$ and $k = 1, \dots, N_S$).

In the coding described above, some problem may arise where the magnitude of the song is very small. The spectral information may contain large amount of uncertainty. So, we take the feature equal to zeros for the segment of the sound whose peak is lower than a threshold, and the feature vector is normalized to sum up to 1.0 for the other segment in order to eliminate the difference caused by the volume of the sound.

Table 1 shows the birds we used in this work.

Table 1. Example of birds' kinds for experiment

bird code	number of vectors	Japanese name	English name
a11-01	343	hoojiro	Siberian Meadow
a11-02	471		Bunting
a27-01	817	mejiro	Japanese
a27-02	773		White-eye
a34-01	687	aoji	Black faced
a34-02	731		Bunting
b11-01	257	koruri	Siberian
b11-02	343		Blue Robin
b24-01	515	nojiko	Japanese
b24-02	429		Yellow Bunting
c04-01	515	kayakuguri	Japanese
c04-02	385		Accentor

3 DP Matching for the Feature Vectors

Among various approaches for matching of similar sequences, DP (dynamic programming) matching is a simple and powerful approach, thus it is often used in speech recognition problems (*e.g.* [1]). Sequences of feature vectors that are some part of spectral density are used for matching.

The bird songs we encounter in parks or woods usually include features such as non-unique repeating times, beginning at the intermediate part, containing abrupt changes, and so on. Hence it is essential to cope with such incompleteness of the song to be identified. Moreover, the spectrum of the corresponding song is not the same to the template because the birds of the test song are not the individuals of the template song.

The following is the result of applying DP matching directly to the sequences of the feature vectors. The silent part (*i.e.* less than the threshold) is eliminated in the sequences before the matching algorithm is applied.

Figure 1 shows the correspondence by DP matching for the same data. This shows the validity of the program.

Figure 2 shows the result for the different song of the same kind of bird (class). As we can see, a good correspondence relation has been obtained. Figures 3-6 show various correspondence relations for the same class or different classes. We can find that for the same class data, the relation is smooth, while for the different class data, the relation is not smooth in general.

We computed DP matching using the sequences of the feature vectors defined in the previous section.

We have shown good results for rather complete song data. However, it is not always the case. Sometimes the birds stop singing suddenly or sing incomplete songs. To check this phenomenon, we intentionally discarded the second half of the song a11.02 and tried DP matching. Figure 7 is the matching result.

Figure 1. Correspondence by DP matching for the same data (a11.1)

Figure 2. Correspondence by DP matching for the same class (a11.1 vs. a11.2)

We can see that the corresponding points have changed. Compare this result with Fig. 2.

4 Clustering of Data

As we showed in the previous section, the data has too much variety to yield a useful result by simply applied DP matching. Based on this fact, we will consider clustering data in order to suppress the unnecessary differences in the data and to make it easy to concentrate on the intrinsic differences.

For this unsupervised learning problem, the Self-Organizing Map (SOM) by Kohonen [3, 4, 5] may be useful.

Thus we use $x(k)$ as the input vector of SOM.

4.1 Structure of SOM

Two-dimensional SOM is a discrete map

$$\{(i, j) \mid i = 1, \dots, p; j = 1, \dots, q\}$$

Figure 3. Correspondence by DP matching for the same class (a11.1 vs. a11.3)

Figure 5. Correspondence by DP matching for the different class (a11.1 vs. b11.1)

Figure 4. Correspondence by DP matching for the different class (a11.1 vs. a27.1)

Figure 6. Correspondence by DP matching for the same class (a27.1 vs. a27.2)

where each grid point (i, j) has a codebook vector $\mathbf{m}_{(i,j)} \in \mathbf{R}^n$. Note that the dimension of $\mathbf{m}$ is equal to that of the input vector $\mathbf{x}$.

The coordinate of the unit is expressed as (i, j) , where $i \in \{1, 2, \dots, p\}$ and $j \in \{1, 2, \dots, q\}$. The codebook vector is a representative of the data, and each instance of data is mapped onto the unit whose codebook is nearest among all the units.

There are some useful features for SOM. First, the clusters have a topological relation among each other by 2-D surface. It is also possible to construct 1-D SOM or 3-D SOM, but 2-D is most useful in the sense that the relation between the clusters can be seen on the screen. Second, the map is self-organized, and it does not require the human's assistance for learning. However, the number of the units or the clusters must be pre-defined before learning.

The size of the SOM ($p \times q$) is an important parameter because the property of SOM is largely dependent on the size. It is necessary to define what we want to get from the map first. Here we will define our objectives as the

followings:

- Recognize the test song as one of the birds shown in Table 1, or
- Neither of them.

Each of the vector to be used for SOM is the spectral density function. By considering the variety of the sound, we will use a map of 10×10 size.

4.2 Training of SOM

As the training data, suppose we have r kinds of birds, and the bird b has n_b training songs.

SOM is trained by the following procedure.

1. Initialize SOM. Randomly generate the codebook vectors $\mathbf{m}_{(i,j)}$, $i = 1, \dots, p$, $j = 1, \dots, q$.
2. Repeat the training by gradually making the neighborhood size small.

Figure 7. Correspondence by DP matching for the same class (a11.1 vs. a11.2) with second half of a11.2 missing

- Pick up the training datum x from the data set.
- Find the unit (i^*, j^*) whose codebook vector is nearest among all the codebook vectors.
- Move the corresponding codebook vector slightly toward x . That is

$$m_{i^*, j^*} := m_{i^*, j^*} + c(x - m_{i^*, j^*})$$

and also train the neighbor units in the same way, where c is a small positive parameter.

5 Recognition Using SOM

The generated SOM can be used for recognizing the song of a bird as one of the six birds or anything else. It should be emphasized that the six kinds of the birds are similar in their songs. They are collected so in order to make a clear distinction between similar songs of birds.

5.1 Song Map

After completing training SOM, it is ready to make distinction between the feature vectors of various songs of the birds. Now we make Song Map as the followings.

Song map has a grid structure $\{(i, j) \mid i = 1, \dots, p; j = 1, \dots, q\}$. Each grid has a vector v of the size $l = \sum_{b=1}^r n_b$. This size is the number of birds used for training SOM.

The vectors $v_{(i, j)}$ ($i = 1, \dots, p; j = 1, \dots, q$) are determined in the following way. On the map after training, all the training vectors are again mapped onto SOM and count the number of the vectors assigned on each unit. On unit (i, j) , suppose the numbers of the vectors came onto this unit be $\nu_j, j = 1, \dots, l$. Also, it is possible to obtain the vectors $w_{(i, j)}$ ($i = 1, \dots, p; j = 1, \dots, q$) that indicate the cases of same kind of birds simply by summing up the same kind of birds. Thus $w_{(i, j)}$ is r -dimensional.

5.2 Evaluation of the Song Map

It is desirable that each of the unit of Song Map has at most one kind of bird to make clear distinction. However, we cannot expect such an ideal case. A criterion of the goodness of the map is given by the entropy given by

$$E = \sum_{i=1}^p \sum_{j=1}^q \frac{P_{(i, j)}}{N} \sum_{b=1}^r \log(\pi_{(i, j)}(b)) \pi_{(i, j)}(b)$$

where

$$P_{(i, j)} = \sum_{b=1}^r w_{(i, j)}(b)$$

$$N = \sum_{i=1}^p \sum_{j=1}^q P_{(i, j)}$$

$$\pi_{(i, j)}(b) = \frac{w_{(i, j)}(b)}{\sum_k w_{(i, j)}(k)}$$

This criterion will be used in the evaluation of the training by SOM. The smaller value indicates a better distinction.

5.3 Recognition

Table 2 shows the mapped number of vectors of the bird 1.

Table 2. The values of Song Map for the song a11-01

	1	2	3	4	5	6	7	8	9	10
1	0	1	1	0	0	3	2	3	2	3
2	0	1	0	0	1	4	2	2	3	0
3	0	1	0	0	0	4	2	0	0	0
4	0	0	0	1	1	3	0	0	0	1
5	0	0	5	1	1	0	1	4	3	3
6	0	0	2	3	4	0	0	0	0	0
7	2	1	5	0	11	0	0	1	1	0
8	1	0	0	0	1	0	0	2	3	4
9	1	1	0	0	0	0	1	1	0	0
10	4	0	0	1	0	2	4	2	0	226

Note that the codebook vector of the unit (10,10) is fixed to all zeros. The attracted vectors to that unit are the ones for the case of less than the threshold.

Although whole the table takes too much space, we show the probability of all the birds at unit (1,1) in Table 4. If a vector falls to this unit, the possibility of the birds a34 or b11 is large.

SOM can be used for pattern classification. In Tanaka et al. [8], a small SOM has been used for pattern classification where the weighting of the elements was used. In

Table 3. Probability of the kind 1 at each grid point

	1	2	3	4	5	6	7	8	9	10
1	0.04	0.03	0.03	0	0.04	0.18	0.12	0.23	0.14	0.11
2	0	0.07	0	0	0.14	0.4	0.14	0.18	0.14	0.08
3	0.02	0.07	0.05	0.06	0.20	0.38	0.33	0.11	0.05	0
4	0	0	0.14	0.26	0.08	0.42	0	0.05	0	0.17
5	0	0.10	0.47	0.25	0.33	0.18	0.33	0.39	0.27	0.14
6	0	0	.42	0.46	0.90	0	0	0	0	0
7	0.11	0.13	0.26	1	0.5	0	0.06	0.15	0.10	0
8	0.24	0.13	0.20	0	0.40	0.17	0.11	0.20	0.23	0.10
9	0.13	0.10	0.10	0.05	0	0.11	0.29	0.10	*	*
10	0.10	0	0	0.05	0.10	0.12	0.38	0.18	*	0.13

Table 4. The contents of the unit (1,1)

a11	0.04
a27	0
a34	0.63
b11	0.20
b24	0.11
c04	0.02

Tanaka and Kajitani [9], SOM was used for visualization of the class of used cars.

Here in this paper, we don't decide the class of each unit deterministically but we use the probabilities of the class of each class.

The possibility of belonging to class b for the feature vector sequence $S = \{x(1), \dots, x(N_S)\}$ is given by the following algorithm.

1. for $k = 1, \dots, N_S$
 - (a) Find the unit (i^*, j^*) that best matches $x(k)$.
 - (b) $C_S(b) := C_S(b) + \pi_{(i^*, j^*)}(b), b = 1, \dots, \tau$

Table 5 shows the highest (1) and the next highest (2) classes for each test data that were not used in training SOM. This result shows that there are some songs difficult to distinguish, although the values among the best and the second ones are not much different in the wrong judgment cases.

6 Conclusions

We first applied DP matching for matching of two songs, and found that the approach was successful for the complete songs. DP does not necessarily work well when the song is not complete. For matching of bad condition songs, we proposed using Self Organizing Map of Kohonen, and proposed a method to evaluate the similarity by accumulating the likelihood of the feature vectors. We are seeking a way to combine SOM and DP

Table 5. Order of the probable classes for each test data

	a11	a27	a34	b11	b24	c04
a11-03	2	1				
a11-04	2	1				
a11-05	1	2				
a11-06	1	2				
a27-03	2	1				
a27-04	2	1				
a27-05		1				2
a27-06	2		1			
a34-03			1	2		
a34-04			1		2	
a34-05			1	2		
a34-06			1		2	
b11-03			1	2		
b11-04			1	2		
b24-03			1		2	
b24-04			2		1	
b24-05			1		2	
b24-06			1		2	
c04-03		2				1
c04-04		1				2
c04-05		2				1
c04-06		2				1

matching so that SOM can deal with the dynamics of the feature vectors.

References

- [1] S. Nakagawa: *Pattern Information Processing*, Maruzen (1999) (in Japanese)
- [2] S. Imai: *Speech Signal Processing*, Morikita Publisher (1996) (in Japanese). T. Kohonen, Analysis

- of a simple self-organizing process, *Biological Cybernetics*, **44** 135/140(1982)
- [3] T. Kohonen, Self-organizing formation of topologically correct feature maps, *Biological Cybernetics*, **43** 59/69(1982)
 - [4] T. Kohonen, *Self Organizing Maps*, Springer-Verlag (1995)
 - [5] T. Kohonen: *Self-Organizing Maps*, Third Edition, Springer (2001)
 - [6] Wild Bird, living in Japan, 2, 3, Columbia (music CD) (1995)
 - [7] A. Iwata, T. Kitamura and Y. Yokota: *Digital Signal Processing*, Corona Publisher Co. Ltd. (1995)
 - [8] M. Tanaka, Y. Furukawa and T. Tanino: Weight tuning and pattern classification by self-organizing map using genetic algorithm, *3rd IEEE Int. Conf. on Evolutionary Computation (ICEC'96)*, 602/605 (1996)
 - [9] M. Tanaka and Y. Kajitani: Interactive weighting SOM for data mining, in N.Baba, C.J. Jain and R.J. Howlett (Eds.) *Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*, IOS Press, 37/41 (2001)