Ensemble Learning with Neural Networks for Classifying Environmental Sounds

Ayako Hiramatsu[1], Asato Simotaki[1], Kazuo Nose[1]
Toshio Minakata[2], Kenji Tennmoku[2], and Osamu Hattori[2]
[1]Department of Information Systems Engineering, Osaka Sangyo University,
3-1-1, Nakagaito, Daitou, Osaka, 574-8530, Japan
ayako@ise.osaka-sandai.ac.jp
[2] Sumitomo Electric Industries, Ltd, 1-1-3, Shimaya, Konohana-ku, Osaka, 554-0024, Japan

ABSTRACT

This paper proposes a classification method for environmental sounds based on neural networks. However, neural networks need trail and error, which are very tedious tasks. To simplify classification accuracy, we investigate two popular ensemble learning methods: Bagging and AdaBoost. We experimentally compare their performances with a single neural network. The results show that their performance is slightly improved and that bagging works more effectively than AdaBoost.

Keywords: Traffic sounds, Ensemble learning, Neural networks, Bagging, Boosting

1. INTRODUCTION

Research in environmental sound classification is less common than research into sound recognition of voice or music etc. This paper discusses a classification method using neural networks for environmental sounds especially traffic sounds. Neural networks are typical classification methods that are often applied in various researches. However, to acquire enough precision, researchers must adjust parameters by trial and error [1][2]. To cope with this problem, we adopt ensemble learning, a method that improves classification by using a group of classifiers [3].

When using neural networks as classifiers for ensemble learning, we expect that classification accuracy will be improved without trial and error for parameters and network constructions. In this paper, therefore, we apply ensemble learning with neural network classifiers to improve classification accuracy.

In the next section, changing from environmental sound data to numerical data for classification is described. In the third section, typical ensemble learning methods (Bagging and AdaBoost) are explained. After experiments that evaluate the efficiency of the proposed method are shown, conclusions are described in the fifth section.

2. ENVIRONMENTAL SOUND DATA

To classify environmental sounds using neural networks, sound data are expressed numerically. Sudden changes of sound pressure are considered triggers. After pulling triggers, sounds are operated by 1024 Fast Fourier Transform (FFT) and features as frequencies are extracted.

To put it concretely, environmental sound data sampled in 48 kHz are divided into sections, as shown in Figure 1. One section includes 1024 samples, and the totals of 1024 samples of sound pressure are regarded as sound pressure differences between sections. If the difference of sound pressure between adjacent sections is bigger than a

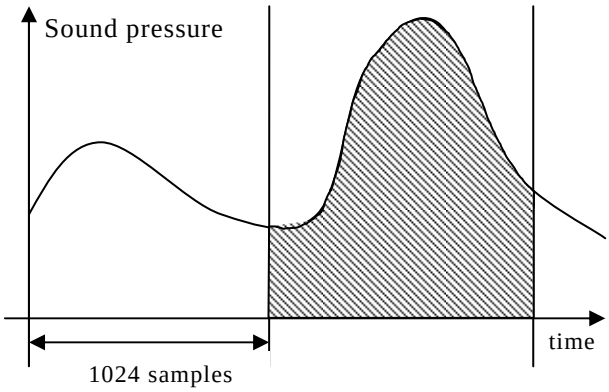

Figure 1: Calculation for sound pressure

certain threshold, the first sample of the section is considered the start of an environmental sound case.

As shown in Figure 2, sound pressure data in the four sections (the start section and the following three sections) are transformed to frequency features by 1024 FFT. This method considers the maximum value among the four sections in each frequency as the basic feature.

Because the basic features of a section consist of 512 values by 1024 FFT, neural networks with basic features become very large. Therefore, the averages of all 16 values are computed and 32 average values are considered the features of a sound case. Moreover, these 32 feature values are normalized within 0-1 to input neural networks.

3. ENSEMBLE LEARNING

The proposed method applies Bagging [4] and AdaBoost [5][6] as typical ensemble learning methods that use combinations of classifiers generated by a certain number of learned weak classifiers (in this paper neural networks are applied).

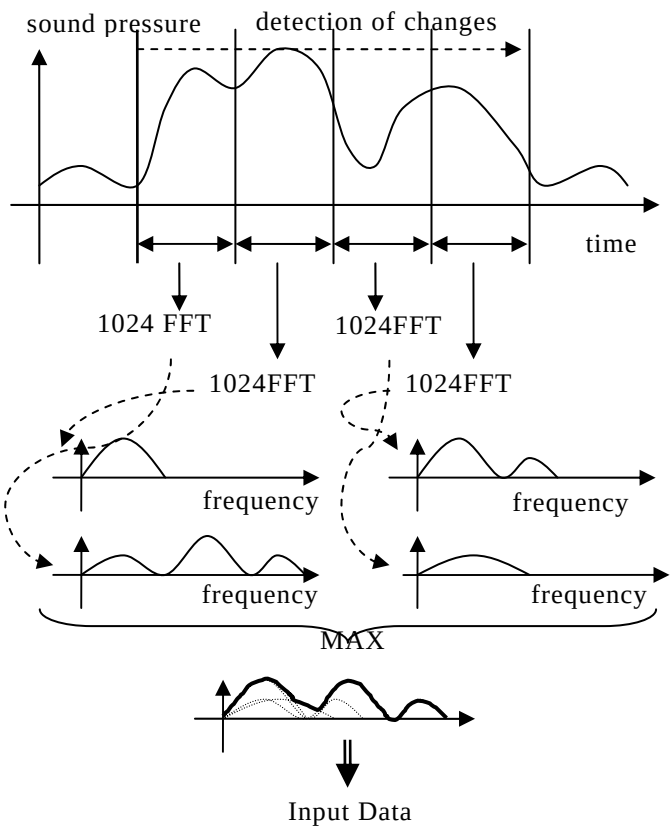

Figure 2: Input data

In the following explanation, $\mathbf{x}_i$ shows a feature vector of input to classifiers, $y_i \in Y = \{1,..,c\}$ indicates a category label of $\mathbf{x}_i$, and a training set $S = \langle (\mathbf{x}_1, y_1),..,(\mathbf{x}_N, y_N) \rangle$ includes N samples.

3.1 Bagging

Bagging method selects N samples randomly based on fixed probabilities ($1/N$) from an original training set consisting of N samples (but

Original Training Set	1, 2, 3, 4, 5, 6
Resampled Training Set 1	2, 3, 1, 1, 4, 5
Resampled Training Set 2	3, 1, 5, 2, 3, 4
Resampled Training Set 3	6, 1, 2, 3, 2, 5

Figure 3: Example of resampled training sets

repeated selection of the same samples is permitted). One classifier is generated from learning with the selected cases.

Figure 3 shows an example of resampled training sets (Bootstrap [7]). In this example, an original training set consists of six samples. In the original training set below, three resampled training sets are expressed.

[Training phase]

- Samples: $\langle(\mathbf{x}_1, y_1),...,(\mathbf{x}_N, y_N)\rangle$
- Label: $y_i \in Y = \{1,...,c\}$

1. Initialize parameters.

- Ensemble: $H_0 = \varnothing$
- Number of classifiers to train: L

2. For $k = 1,..,L$

- Take a bootstrap [7] sample S_k from S.
- Build a classifier h_k using S_k as the training set.
- Add classifier to current ensemble:

$$H_k = H_{k-1} \cup h_k$$

[Classification phase]

3. Run $h_1,...,h_L$ on input $\mathbf{x}$.
4. Class with maximum number of votes is chosen as label for $\mathbf{x}$.

Figure 4: Bagging algorithm

As shown in this figure, some samples such as 1,2, and 3 may be selected repeatedly, but other samples such as 4,5, and 6 are selected only once or not at all.

In the Bagging method, generation of classifiers by random sampling and learning are repeated to improve classification performance using a group of generated classifiers. The Bagging method algorithm is shown in Figure 4.

In Breiman's original Bagging method [4], a final classification result is decided by the most approvable category for which each learned classifier votes. However, because the output of neural networks is a continuous value, how neural networks decide one category for one vote based on such values is problematic. Moreover, output to categories except the voted category affects nothing. In this paper, the output of each classifier is totaled, and a final Bagging classification result is the category that has the best value [8] (Formula (1)).

$$h_{fin}(x) = \arg\max_{y \in Y} \sum_{k=1}^{L} h_k(x, y) \qquad (1)$$

3.2 AdaBoost

To apply AdaBoost to neural networks, we use Freund's Adaboost.M2 algorithm, as shown in Figure 5. B in Figure 5 is a set of all mislabels $B = \{(i, y) : i \in \{1,..,N\}, y \neq y_i\}$, i is an index of sample $\mathbf{x}_i$, and y is a pair of incorrect labels (i, y) for sample $\mathbf{x}_i$. $|B|$ shows the number of factors, which here is $|B| = N(c-1)$.

[Training phase]

- Samples: $\langle(\mathbf{x}_1, y_1),...,(\mathbf{x}_N, y_N)\rangle$,
- Label: $y_i \in Y = \{1,...,c\}$

1. Initialize parameters.

- Ensemble: $H_0 = \varnothing$
- Number of classifiers to train: L
- $w_1(i,y) = 1/|B|$ for $(i,y) \in B$

2. For $k = 1,...,L$

- Take sample S_k from S using mislabel distribution w_k.
- Build classifier h_k using S_k as the training set.
- Calculate the pseudo-loss of h_k:

$$\varepsilon_k = \frac{1}{2} \sum_{(i,y)\in B} w_k(i,y)(1 - h_k(x_i, y_i) + h_k(x_i, y))$$

- Set $\beta_k = \varepsilon_k / (1-\varepsilon_k)$.
- Update w_k:

$$w_{k+1}(i,y) = \frac{w_k(i,y)}{Z_k} \cdot \beta_k^{(1/2)(1+h_k(x_i,y_i)-h_k(x_i,y))}$$

where Z_k is a normalization constant (chosen so that w_{k+1} will be a distribution).

- Add the classifier to the current ensemble:

$$H_k = H_{k-1} \cup h_k$$

[Classification phase]

3. Run $h_1,..., h_L$ on input $\mathbf{x}$.

4. $$h_{fin}(x) = \arg\max_{y\in Y} \sum_{k=1}^{L} (\log \frac{1}{\beta_k}) h_k(x,y)$$

Figure 5: AdaBoost.M2 algorithm

The AdaBoost method gradually changes sampling rates w to choose learning cases every time one classifier h_k is generated. For this procedure, difficult cases are learned more and classifiers that can deal with difficult cases are gradually generated.

Figure 6 is an example of resampled training sets by AdaBoost. This original training set consists of six samples. Here, suppose that sample 6 is difficult to learn. From this example, it is understandable that sample 6 will be chosen much later.

Original Training Set	1, 2, 3, 4, 5, 6
Resampled Training Set 1	6, 3, 2, 5, 4, 2
Resampled Training Set 2	6, 5, 1, 2, 1, 6
Resampled Training Set 3	6, 6, 2, 1, 6, 6

Figure 6: Example of resampled training sets

4. EXPERIMENTS

4.1 Target data

Table 1 shows targets of environmental sound data of classification, which include 11 categories and 119 cases.

Each category shows a certain sound source that belongs either to group A or B. Groups are defined based on a point of view that a sound is either an emergency case or a normal case.

In numerical experiments, we compared three methods (single NN, Bagging, and AdaBoost) from three points: classification accuracy, parameter adjustment, and number of classifiers.

Table 1: Sound samples

Group	Category	Number of Samples
Group A	N1	10
	N2	8
	N3	29
Group B	N4	6
	N5	6
	N6	8
	N7	15
	N8	10
	N9	12
	N10	6
	N11	9

4.2 Comparison of classification accuracy

We did experiments on both category and group classification. We also tried an experiment that used categories in the learning phase and groups in the classification phase. Moreover, we compared three methods: simple neural networks, Bagging with neural networks, and AdaBoost with neural networks. To compare the three methods, we examined 20 six-fold cross-validations for each experiment. In six-fold cross-validation, cases are divided into six groups. Five groups are used to train and one group is used for tests. Because we used 119 cases, about 20 cases were included in a group. In Bagging and AdaBoost, 10 classifiers are generated. The parameters of neural networks in all experiments are shown in Table 2.

Table 2: Parameters for neural networks

Number of Input Units	32
Number of Output Units	11 in Category Level 2 in Group Level
Number of Hidden Layers	1
Number of Hidden Units	20
Learning Rate	0.2
Range of Outputs	0～1

Results of experiments are shown in Table 3 and show classification accuracy by simple neural networks (NN), Bagging, and AdaBoost as averages of 20 trials.

Table 3: Classification Results

	NN	Bagging	AdaBoost
Category Level	0.554	0.572	0.556
Group Level	0.766	0.804	0.793
Learned with Category Level, then Classified with Group Level	0.800	0.808	0.805

The experiments clarified the following points.

- With category level experiments, accuracy rates are from 55% to 57% and Bagging is the best but differences between the three methods are very small.
- With group level experiments, accuracy rates are from 77% to 80%. Bagging with neural networks is the best.
- With experiments learned with category level, then classified with group level, the accuracy rates of the three methods are about 80% with no clear differences.
- In all experiments, there are only a few differences, but the Bagging method with neural networks can classify accurately.

4.3 Influence of parameter adjustments

Parameter adjustment is necessary for neural networks. We examined how effective ensemble learning prevents troublesome parameter adjustments. This experiment was executed with category levels in both the training and classification phases.

For preparation, we examined classifications with various parameters by a single neural network as parameter adjustment steps. For these results, Table 4 shows the best and the worst two parameter combinations.

Table 4: Parameter combinations

	Best accuracy		Worst accuracy	
Number of Hidden Units	10	20	10	20
Learning Rate	0.35	0.15	0.05	0.05

With each condition shown in Table 4, we examined classification experiments by a single neural network, Bagging with neural networks, and AdaBoost with neural networks. Accuracy rates are calculated by averaging 10 six-fold cross-validation results. The number of classifiers is 10. The other conditions are the same as the experiments in section 4.2. Results are compared in Table 5.

Although in a single neural network, about 15% accuracy difference occurs by quality of parameters, Bagging is about 10% and AdaBoost is only 7%. Therefore, when using Bagging and AdaBoost, it is not necessary to adjust parameters as strictly as single neural networks.

Table 5: Comparison of results in different parameters

Learning Rate	0.05		0.15	0.35
Number of Hidden Units	10	20	20	10
NN	0.405	0.449	0.558	0.536
Bagging	0.450	0.501	0.538	0.559
AdaBoost	0.487	0.514	0.552	0.542

4.4 Number of classifier experiments

We examined whether the number of classifiers influences classification accuracy. Accuracy rates are calculated by averaging 10 six-fold cross-validation results. As experiment conditions, the number of hidden units is 10, the learning rate is 0.35, and other neural network parameters are the same as the experiments in section 4.2. However, the number of classifiers is changed from 1 to 100. Results are shown in Figure 7.

In 5–20 classifiers, AdaBoost has the best accuracy, and after 20 classifiers Bagging has the best. With more than 50 classifiers, accuracy rates sometimes exceed 60%. However, an improvement trend is not clearly observed.

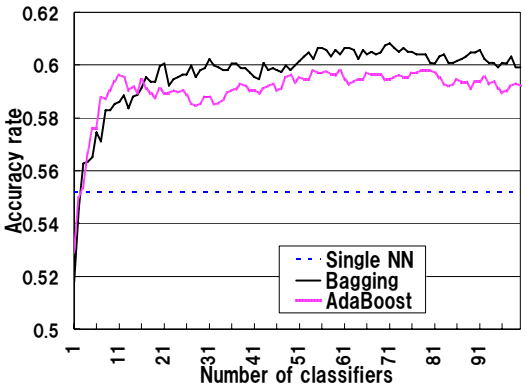

Figure 7: Change of accuracy rate by number of classifiers

5. CONCLUSION

This paper described a method that classifies environmental sounds especially traffic sounds. To improve classification accuracy, we applied Bagging and AdaBoost, which are typical ensemble learning methods. Experiment results indicate that ensemble learning methods are superior to single neural networks in accuracy rates and parameter adjustments.

REFERENCES

[1] Russell D. Reed and Robert J. Marks II: "Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks," MIT Press (1999)

[2] Sarle, W.S., ed: "Neural Network FAQ," ftp://ftp.sas.com/pub/neural/FAQ.html (1997)

[3] Ludmila I. Kuncheva: “Combining Pattern Classifiers: Methods and Algorithms,” Wiley (2004)
[4] Leo Breiman: “Bagging Predictors,” Technical Report 421, Department of Statistics, Univ. of California at Berkeley (1994)
[5] Yoav Freund and Robert E. Schapire: “Experiments with a New Boosting Algorithm,” International Conference on Machine Learning, pp. 148-156 (1996)
[6] Yoav Freund and Robert E. Schapire: “A Decision-theoretic Generalization of On-line Learning and an Application to Boosting,” Journal of Computer and System Sciences, 55(1):119-139 (1997)
[7] Bradley Efron and Robert J. Tibshirani: “An Introduction to the Bootstrap,” Chapman & Hall/CRC (1993)
[8] Harris Drucker: “Boosting Using Neural Networks,” Combining Artificial Neural Nets: Ensemble and Modular Learning, Amanda J. C. Sharkey (ed), pp. 51-77, Springer (1999)