# Brain Computer Interface
# using Silent Speech for Speech Assistive Device

Mariko Matsumoto

Doctoral Program in Information Science and Engineering

Graduate School of Science and Technology

Niigata University

**Contents**

# 1. Introduction

## 1.1. Background

Daily life demands that we use verbal and non-verbal communication. However, severely handicapped individuals such as people with advanced amyotrophic lateral sclerosis (ALS), locked-in syndrome, or nasopharyngeal cancer have difficulty expressing their thoughts. Their caregivers also face difficulties when caring for patients. A brain–computer interface (BCI) has been developed to provide prosthetics for such individuals.

The anticipated benefits of the BCI are not merely confined to those individuals. They are expected to be useful for entertainment, personal communication, and game devices, and with preventive medical treatments for healthy individuals. When both healthy individuals and those with a disability use the same core technology, the demand shown by healthy people is expected to contribute to the welfare of handicapped individuals through improved production and reduced costs of assistive equipment. I seek to develop devices that are attractive for both handicapped and healthy people. The objective technology requires portability, high classification accuracy, and usability.

For these supporting prosthetics, many studies have been conducted using methods such as P300 speller [1], steady-state visual evoked potentials (SSVEP) speller [2], SSVEP cursor controller [3], and near infrared spectroscopy (NIRS) [4]. The P300 needs long period to detection for oddball task and averaging. For the SSVEP spellers, users must gaze on the attempted word. With the SSVEP cursor controller, subjects must undergo training to move cursors using electroencephalography (EEG). In the method using hemodynamic response, e.g., NIRS, users must train in the mode of imagining calculations or imagining fast songs for detection. The methods described above necessitate training of skills that users have never developed in daily life.

The classification of silent speech is a simple method that requires no special training. In addition, spatial filtering enables silent speech to be detected by single trial. Many silent speech interface studies have used electromyographic (EMG) signals [5], electromagnetic field measurements with implanted magnets [6], and ECoG signals detected using invasive electrodes [7]. The method using EMG signals requires electrodes mounted on the user's face or neck. The system is uncomfortable and fragile. The method using magnet implantation around a patient's mouth is effective, but it

requires surgical operations. Severely paralyzed patients might accept surgical operations, but healthy individuals would not accept them. Moreover, any method using invasive electrodes necessitates surgical operations. The detection of silent speech by EEG is a good method in terms of portability and user-friendliness. It doesn't need surgical operation. In addition, headset device which detects EEG is hard to break.

I recommend that method using silent speech and EEG is the best method to help communication (Fig. 1.1).

The imagined vowels were classified using EEG, as measured using scalp electrodes, common spatial pattern (CSP) filtering, and nonlinear support vector machine (SVM) [8]. The CSP method is commonly used to find spatial filters for the classification of multichannel EEG signals. The spatial filters for multichannel EEG signals, which are derived using CSP, can extract discriminatory information from two classes of EEG signals and enables to be detected by single trial. The SVM separates two classes with maximized margin nonlinearly. Classification rates of 56–72% were obtained for 64 electrodes for the pairwise classification /a/ vs. /u/. This classification rate is insufficient for feasibility.

With regard to detecting the imagined voice, some problems arose: the related brain geometries and suitable electrodes for classifications differed among subjects. For that reason, I used adaptive collection which divided signals after CSP filtering into small elements and evaluated them relative to the elements. It thereby selected the better elements for classification.


**Figure 1.1**. Assistive BCI for speech

The imagined vowels were classified using EEG, as measured using scalp electrodes, common spatial pattern (CSP) filtering, and nonlinear support vector machine (SVM) [8]. The CSP method is commonly used to find spatial filters for the classification of multichannel EEG signals. The spatial filters for multichannel EEG signals, which are derived using CSP, can extract discriminatory information from two classes of EEG signals and enables to be detected by single trial. The SVM separates two classes with maximized margin nonlinearly. Classification rates of 56–72% were obtained for 64 electrodes for the pairwise classification /a/ vs. /u/. This classification rate is insufficient for feasibility.

With regard to detecting the imagined voice, some problems arose: the related brain geometries and suitable electrodes for classifications differed among subjects. For that reason, I used adaptive collection which divided signals after CSP filtering into small elements and evaluated them relative to the elements. It thereby selected the better elements for classification.

### 1.2. Purpose of thesis

The overall aim of this thesis is to show the feasibility of speech assistive BCI using silent speech.

Silent speech is that a subject imagines vocalization while he/she remains silent and immobilized. The benefit is that critically ill patients who can't vocalize by themselves, can make silent speech and it doesn't need special training and single trial detection is possible.

My research started from vowels because Japanese syllables are based on five vowels, /a/, /i/, /u/, /e/, and /o/. Most Japanese syllables consist of one of five vowels and a consonant. Therefore, I studied vowels at first.

Regarding feasibility, some problems arose: The classification accuracies were insufficient. The large number of electrodes is un-convenient. Multi-class classification and online processing are required.

The SVMs are well known as a pairwise classifier. For online processing, classification speed is important and large calculation cost is problem, but SVM with Gaussian kernel (SVM-G) has hyperparameters that must be optimized by cross validations.

The relevance vector machine (RVM) was proposed as a method, which has small number of relevance vectors and optimize the hyperparameters automatically [22, 23].

In my first paper [9], I proposed adaptive collection to increase the classification accuracies and showed the feasibility of the assistive BCI for speech.

In my second paper [10], as preparation for the feasibility study, I compared RVM and SVM to search better algorithm in terms of calculation cost.

This thesis is a combination of the previous two papers above.

### 1.3. Outline

Fig. 1.2 outlines this thesis, which consists introduction (chapter 1), experiments (chapter 2), methods (chapter 3), results (chapter 4), discussion (chapter 5), and conclusion (chapter 6). The method includes adaptive collection, which I proposed.

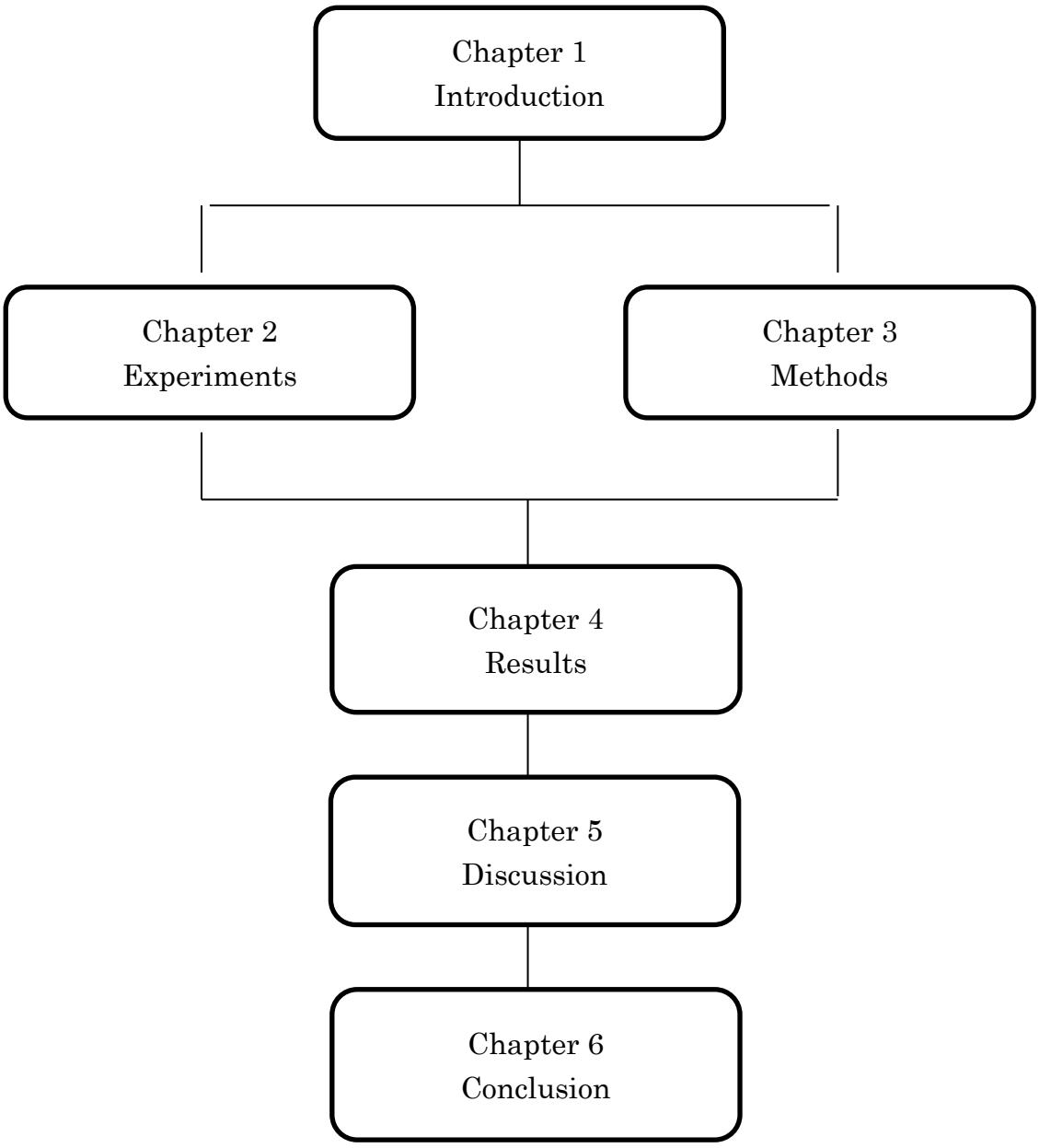



**Figure 1.2**. Outline of this thesis

## 2. Experiments

### 2.1. Subjects

The experiments involved five 21–24-year-old male participants (S1–S5). All subjects were native speakers of Japanese who were right-handed, as assessed by the Edinburgh Inventory [11]. No participant had any neurological disorder or noteworthy health problem. Experiments were conducted in accordance with the Declaration of Helsinki. Informed consent was obtained from all subjects.

### 2.2. Experiments

Each subject was seated comfortably in an armchair with eyes closed to avoid the influence of visual activation. The subjects were coached beforehand and had rehearsed with actual movements a few times to ensure correct task execution. The subjects were then asked to imagine voice production for one second, while remaining silent and immobilized. The Japanese vowels, /a/, /i/, /u/, /e/, and /o/ were imagined. Two tasks were conducted: the fixed order task and the random order task. The tasks used sound commands generated by a personal stereo device (Walkman NW-E053; Sony Corp.). Subjects, while hearing them through earphones, were instructed to perform the following tasks.

**1) Fixed order task**

The timings of onset for imagined vocalization in order were organized in the following manner.

**Task**: Subjects were instructed to imagine the voice production (imagined vocalization) of one of vowels /a/, /i/, /u/, /e/, and /o/ in fixed order for one second following one second for rest. The onset and ending of the imagined vocalization were signaled to the subjects using clicking sounds (Fig. 2.1(a)).

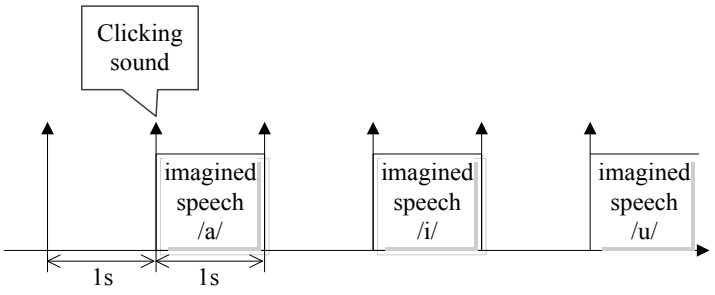
One trial stream consisted of 5 vowels × 13 times for about 2.2 min. Subjects 1–4 performed the experimental set four times. In all, 52 epochs were obtained for each vowel. Consequently, 260 trials were obtained for each subject. I designate this batch of data as 260 epochs. For subject 5, 65 epochs were obtained for each vowel from five experimental sets.

**2) Random order task**

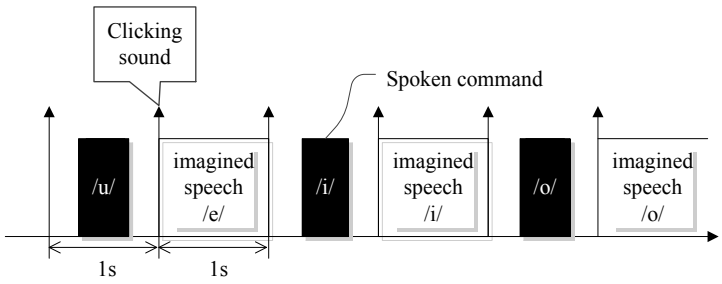
The timings of onset for imagined vocalization were organized in the following manner.

**Task**: Subjects were instructed to imagine the voice production (imagined vocalization) of a vowel that was the same as the last spoken command for one second following one second for rest. The spoken command expressed one of the vowels, /a/, /i/, /u/, /e/, or /o/ in randomized order. The onset and ending of the imagined vocalization were signaled using clicking sounds (Fig. 2.1(b)).

To avoid the influence of auditory evoked potentials, the interval between spoken commands and onset of the imagined speech was set to 200 ms or more. In all, 50, 45, 52, 52, and 65 epochs were obtained, respectively, for each vowel for subjects 1, 2, 3, 4, and 5. To clarify, in task /a/, for instance, subjects imagined speech production of /a/ for one second, while they remained silent and immobilized. The ways for other vowels are as the same as above.




(a) Fixed order task




(b) Random order task

Figure 2.1. Experimental protocols

### 2.3. Recording

EEG signals were recorded using an electroencephalograph (Neurofax EEG-1100; Nihon Kohden Corp.) and 128 channel Modular EEG-Recording Caps (Easy Cap) [12] with a sampling rate of 1000 Hz. The electrodes were set except for those from 109 to 114 and from 117 to 128 in [12]. To calculate the feature for 63 electrodes, BioSemi B.V. [13] was used to select the 64 EEG positions from the 111 electrodes. To calculate the features for 19 electrodes, international 10–20 [14] was used to select the 19 channel electrodes from the 111 electrodes. One of the 20, 64, or 112 electrodes was used to reduce the humming noise; the remaining electrodes for calculation were therefore 19, 63, and 111 electrodes, respectively. For reference, two electrodes were attached on the right and left ears. One electrode was set below an eye to detect unwanted eye movement and artifacts. However, no artifact rejection algorithm was used for this study.

### 2.4. Data processing

Data processing was performed using software (MATLAB; The MathWorks Inc., Natick, MA). Using a decimation filter with cutoff frequency of 125 Hz, the recorded EEG data were decimated from 1000 Hz sampling to 250 Hz sampling after filtering. 125Hz was determined from 250Hz of sampling frequency by the Nyquist theorem. Epochs were extracted in reference to the stimulus onset. The duration was one second. 52 epochs were extracted for each vowel, the fixed order task, and Subjects 1–4, for a total of 260 epochs per subject. 65 epochs were obtained for each vowel for Subjects 5. 50, 45, 52, 52, and 65 epochs were respectively extracted for each vowel and random order tasks for Subject 1, 2, 3, 4, and 5.

## 3. Method

As described in this paper, I compared three methods, support vector machine with Gaussian kernel (SVM-G), relevance vector machine with Gaussian kernel (RVM-G), and linear relevance vector machine (RVM-L).

Signal processing consists of adaptive collection and common special pattern filters (Fig. 3.1).

### 3.1. Adaptive Collection

Adaptive collection (AC) enables the use of suitable time duration of signals after CSP filtering for classification.

I proposed adaptive collection that adaptively uses better output signals of CSPs and its time durations for classification to improve classification accuracies (Figs. 3.1 and 3.2).

This method derives from frequency allocation techniques in the communication field as my experience. The mobile telephone system partitions wireless resources by time and frequency, and evaluates them by the signal to noise ratio for allocation of effective data transmission. Similarly, adaptive collection divides data into small elements. Each element consists of small time duration and an output signal of a CSP, and evaluates them and uses effective elements for classification by evaluation results.

The AC consists of the t-element generation, the element generation, the evaluation, and the combination and decision (Fig. 3.1).

Two epochs are used as test data and the remaining epochs are divided evenly into training data and validation data every an iteration. Then various combinations of test data, training data, and validation data are used iteratively. Eventually all epochs are used as test data. The validation data are used for evaluation whereas test data are used for classification. For example, when the number of epochs is 52 for each vowel, number of test data is two, number of training data and evaluation data is 51 for pairwise classification.

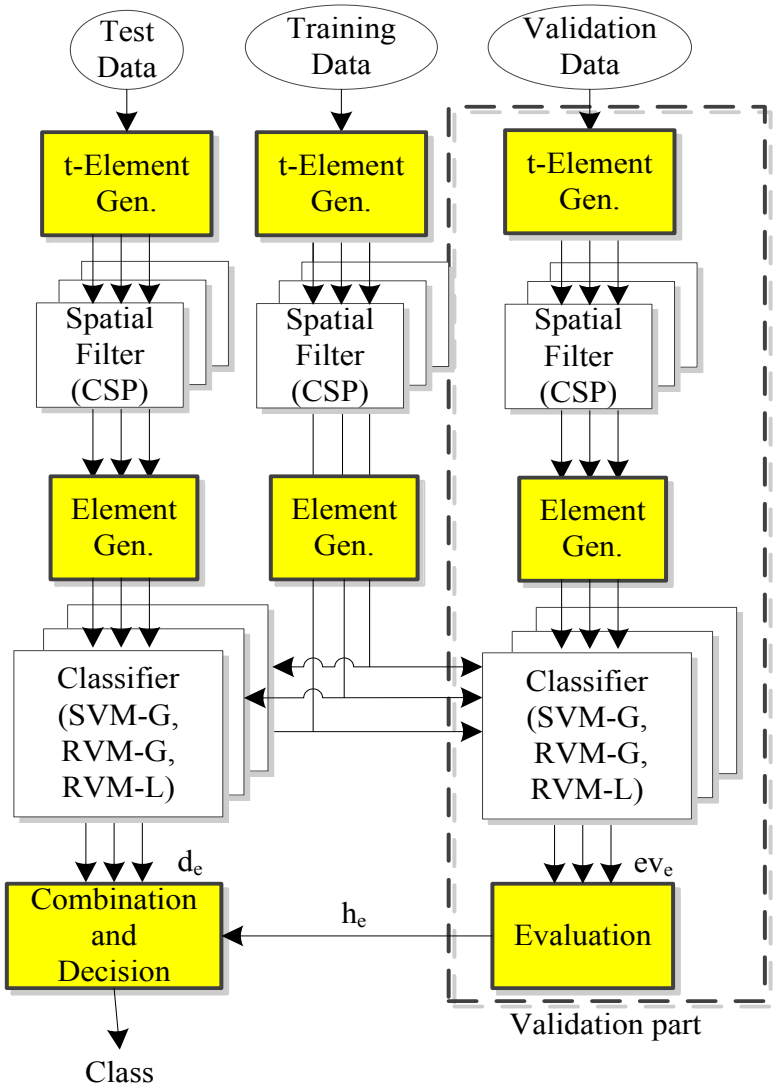



**Figure 3.1** Flow diagram

**1) t-Elements Generation**

First, a low-pass filter with cutoff frequency of 125 Hz is applied to change sampling speed to 250. Second, data is divided in the time domain into t-elements. Each t-element consists of all channel signals with time duration of *Tt*, which is set to 25 samples (100 ms). The t-element label is *t-e*=( *t*), where *t* denotes time label ( *t*=1, 2, ⋯, 10 ) (Fig. 3.2). The t-elements are made for application of CSPs.

**2) Element Generation**

After CSPs, each data is divided into elements. Each element consists of one output signal of a CSP with time duration of *Tt*. The element label is *e*=(*s*, *t*), where *s* denotes signal label (*s*=1, 2, ⋯, 63) (Fig. 3.2).

**3) Evaluation**

To ascertain the suitable elements for classification, I evaluate the performance related to the elements per subject and vowel combinations. The evaluation uses the validation data and the training data, and these are strictly isolated from the test data. The reliability coefficient $h_e$ was set to one when its classification accuracy $ev_e$ was included in the top *M*, and is otherwise zero. *M* is set to 20.

**4) Combination and Decision**

To use suitable elements from the evaluation results, the combination and detection part outputs the class using the following formula.

$$\text{class} = f\left(\frac{\sum_{e=1}^{N_e} h_e r_e}{\sum_{e=1}^{N_e} h_e}\right) \tag{3.1}$$

Therein, $r_e$ is the classification result related to element *e*, e.g., class 1 or 2, $N_e$ is the total number of elements, and $f(\cdot)$ is the decision function. The parenthesis above represents the average of classification results related to the top *M* elements.

As a result, AC enables the use of suitable time duration of signals after CSP filtering for classification. In other words, AC selects suitable spatial feature of brainwaves for classification because the output signals of CSP are related to eigenvectors.

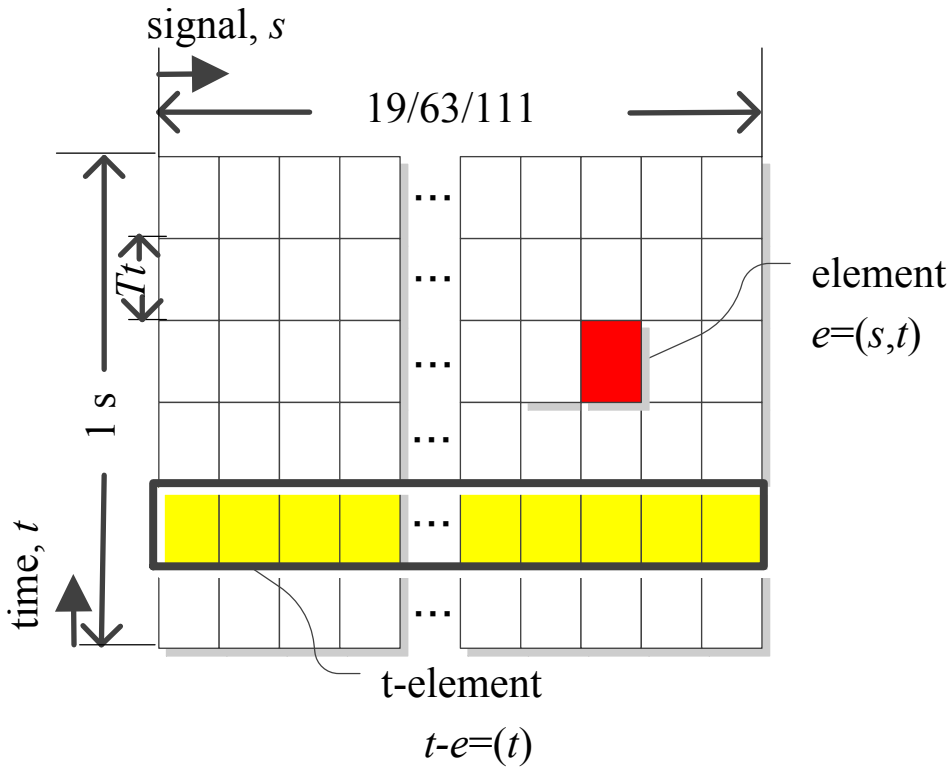



**Figure 3.2** Element Generation

### 3.2. Common Spatial Patterns (CSP)

Using CSP, I designed spatial filters that treat EEG data in a maximally discriminative manner. Using a spatial filter, which spatially gather signals, enables single trial detection without averaging of multiple trials [8]. Detailed descriptions of the CSP method with equivalent equations can be found in reports of studies by Müller-Gerking et al. [15] and Ramoser et al. [16]. To describe them briefly, given two groups of EEG time series data (e.g., tasks to classify /a/ and /u/), I designate each epoch as a matrix $E_g^{t,i}$ in which the rows and columns of *E* respectively denote electrodes and samples, *t* is the time label, *i* is the epoch label, and *g* is the group label. I then compute normalized covariance matrices $C_g^t$ for the epochs of each group and each element and average them such that

$$\boldsymbol{C_g^t} = \frac{\mathbf{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}} \frac{\boldsymbol{E_g^{t,i}}\left(\boldsymbol{E_g^{t,i}}\right)^{\boldsymbol{T}}}{\boldsymbol{trace}\left(\boldsymbol{E_g^{t,i}}\left(\boldsymbol{E_g^{t,i}}\right)^{\boldsymbol{T}}\right)}, \tag{3.2}$$

where *m* is the number of trials in group *g*. The two resultant matrices are summed to produce a composite covariance matrix $C_c^t$, which is then factored into its eigenvectors such that the following apply.

$$\boldsymbol{C_c^t} = \boldsymbol{C_1^t} + \boldsymbol{C_2^t} \tag{3.3}$$

$$\boldsymbol{C_c^t} = \boldsymbol{V_c^t \lambda_c^t V_c^{t^T}} \tag{3.4}$$

Therein, $V_c^t$ is a matrix of eigenvectors. $\lambda_c^t$ is a diagonal matrix of eigenvalues. I then calculate a linear transformation called a "whitening transformation".

$$\mathbf{W^t} = \sqrt{\boldsymbol{\lambda_c^{t^{-1}}}} \boldsymbol{V_c^{t^T}} \tag{3.5}$$

It equalizes the variances in eigenspace. The whitening transformation is then applied to the original two covariance matrices.

$$\boldsymbol{S_g^t} = \boldsymbol{W^t C_g^t W^{t^T}} \tag{3.6}$$

$$\boldsymbol{S_1^t} = \boldsymbol{U^t \lambda_1^t U^{t^T}} \tag{3.7}$$

Thereby, the transformation renders their eigenvectors $U^t$ equivalent and their eigenvalues $\lambda_1^t$ and $\lambda_2^t$ summing to 1, with the diagonal elements of 1 ordered in ascending order. Finally, I define a projection matrix $P^t = \left(U^{t^T} W^t\right)^T$, where the columns of $P^{t^{-1}}$ are the common spatial patterns. They can be regarded as time-invariant EEG source distribution vectors during an element; then I decompose each EEG epoch such that

$$\boldsymbol{Z_g^{t,p} = P^t E_g^{t,p} \quad (p \not\subset i)} \ . \tag{3.8}$$

The resultant feature vectors of $Z_g^{t,p}$ are optimized for discrimination of the two groups, where *p*, the epoch label for the test data, is isolated from *i* of Eq. (3.1). As presented in Fig. 3.1, the exception is only itself for test data. The exception is test data and itself for validation data. In that way, I calculated CSPs for each subject, vowel combination, and time label and used them.

In Fig. 3.1, the input of the spatial filter (CSP) is $E_g^{t,p}$ in Eq. (3.8) and $E_g^{t,i}$ in Eq. (3.1) and the output is $Z_g^{t,p}$ in Eq. (3.8).

### 3.3. Support Vector Machine with Gaussian kernel (SVM-G)

Rakotomamonjy [17] described SVM well. The support vector machine classifier is a binary classifier algorithm that seeks an optimal hyperplane as a decision function in a high-dimensional space [18] . One has a training dataset $\{x_n, y_n\} \in \mathbb{R}^N \times \{-1,1\}$ where $x_n$ are training examples and $y_n$ are the class labels. The method first maps *x* into a high-dimensional space via a function $\emptyset$. Then it computes a decision function of the following form.

$$\boldsymbol{f(x) = \mathbf{w}^T \Phi(x) + b} \tag{3.9}$$

Then, the distance between the set of points $\emptyset(x_k)$ to the hyperplane parameterized by (*w*; *b*) is maximized, while being consistent on the training set. *b* is the bias. The class label of *x* is obtained by considering the sign of *f*(*x*). For the SVM classifier with misclassified examples being quadratically penalized, this optimization problem can be written as

$$\mathbf{min}_{w,\xi}\frac{1}{2}\|\boldsymbol{w}\|^2 + \mathbf{C}\sum_{n=1}^{N}\boldsymbol{\xi}_k^2 \tag{3.10}$$

under the constraint $\forall n,\ y_n f(x_n) \geq 1 - \xi_n$. Therein, $\boldsymbol{w}$ represents the weight vector, $N$ stands for the vector length, $\xi_k$ represents slack variables needed to allow misclassification, and $C$ is a tuning hyperparameter, which controls the tradeoff between maximization of the margin width and minimization of the number of misclassified data in the training set. The solution of this problem is obtained using the Lagrangian theory. One can prove that vector **w** is of the following form.

$$\mathrm{w} = \sum_{n=1}^{N}\alpha_n^* y_n \Phi(x_n) \tag{3.11}$$

Therein, $\alpha_n^*$ is the solution of the following quadratic optimization problem.

$$\max_{\alpha} W(\alpha) = \sum_{n=1}^{N}\alpha_n - \frac{1}{2}\sum_{n,l}^{N}\alpha_n\,\alpha_l y_n y_l\left(K(x_n, x_l) + \frac{1}{C}\delta_{n,l}\right) \tag{3.12}$$

That equation is subject to $\sum_{n=1}^{N} y_n\alpha_n = 0$ and $\forall n$, $\alpha_n \geq 0$, where $\delta_{n,l}$ is Kronecker's delta and $\mathrm{K}(x_n, x_l) = \langle\phi(x_n, x_l)\rangle$ is the Gram matrix of the training examples.

The SVM adopting a radial basis function (RBF) with Gauss kernel is used widely for brain signal classification, as described by Asano et al. [19]. The kernel function of support vectors $x_j$ is

$$\mathrm{K}(x, x_j) = e^{-\frac{\|x - x_j\|^2}{2\sigma}}, \tag{3.13}$$

where $\sigma$ is a parameter related to variation of the training data. Using "SVM and Kernel Methods Matlab Tool box" of an SVM software package, I applied SVMs with Gaussian kernels (SVM-G) for pairwise classification. Parameter $\sigma$ is determined through a grid search and cross-validation of the validation data. Hyperparameter $C$ is determined as heuristic and set to fixed value to avoid increase of calculation cost because $C$ was not so sensitive to the performance. Hsu et al. [20] , used rough tuning to avoid overlearning. The epochs are of three groups: test data, training data, and validation data. Two trial epochs are used as the test data. The remaining epochs are

divided evenly into training data and the validation data. Various combinations are used iteratively. All epochs are used as test data.

In Fig. 3.1, when classifier blocks act as SVM-G classifiers, the input is $x$ and $x_j$ in Eq. (3.13) and the output is classification results that are derived from Eq. (3.9).

### 3.4. Relevance Vector Machine with Gaussian kernel (RVM-G) and Linear Relevance Vector Machine (RVM-L)

The RVM introduces a priori over the model weights governed by a set of hyper-parameters, in a probabilistic framework. Each hyper-parameter is associated with each weight. The most probable values are iteratively estimated automatically. The most compelling feature of the RVM is that it typically uses considerably sparser weighting than SVM, while providing similar performance.

For two-class classification, any target is classifiable into two classes such that $y_n \in \{0,1\}$. A probabilistic distribution can be adopted for $p(t \mid x)$ in the probabilistic framework because only two classes (0 and 1) are possible. The logistic sigmoid link function

$$\sigma(f) = 1/\big(1 + exp(-f)\big) \tag{3.14}$$

is applied to

$$f(x) = \mathbf{w}^T \boldsymbol{\Phi}(\boldsymbol{x}) + b \tag{3.15}$$

to link random and systematic components, and to generalize the linear model. The likelihood is written as

$$p(\boldsymbol{y}|\boldsymbol{w}) = \prod_{n=1}^{N} \sigma\{f(\boldsymbol{x}_n; \boldsymbol{w})\}^{y_n} \left(1 - \sigma\{f(\boldsymbol{x}_n; \boldsymbol{w})\}\right)^{1-y_n} \tag{3.16}$$

for targets $y_n \in \{0,1\}$.

$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots \alpha_n)^T$ shows the hyperparameter introduced to control the strength of the prior over its associated weight. Therefore, the prior is Gaussian, but conditioned on$\alpha$. For a certain value $\alpha$, the posterior weight distribution conditioned on the data is obtainable using Bayes' rule.

The weight cannot be analytically obtained. Therefore, an approximation procedure is used.

1) Because $p(\boldsymbol{w}|y,\boldsymbol{\alpha})$ is linearly proportional to $p(\boldsymbol{y}|\boldsymbol{w}) \times p(\boldsymbol{w},\boldsymbol{\alpha})$, it is possible to find the maximum of

$$\log\{p(\boldsymbol{y}|\boldsymbol{w})p(\boldsymbol{w}|\alpha)\} = \sum_{n=1}^{N}[y_n \log f_n + (1-y_n)\log(1-f_n)]\frac{1}{2}\boldsymbol{w}^T\boldsymbol{A}\boldsymbol{w} \quad , \quad (3.17)$$

where $p(\boldsymbol{y}|\boldsymbol{w})$ is the likelihood of *t*, and $p(\boldsymbol{w}|\boldsymbol{\alpha})$ is the prior density of **w**. For the most probable weight $\boldsymbol{w}_{MP}$ with $y_n = \sigma\{f(x_n;\boldsymbol{w})\}$ and $\mathbf{A} = diag(\alpha_0, \alpha_1, \cdots, \alpha_N)$ being composed of the current values of α. This penalized logistic log-likelihood function requires iterative maximization.

The iteratively reweighed least-squares algorithm is useful to find $\boldsymbol{w}_{MP}$ [22, 23].

2) The logistic log-likelihood function can be differentiated twice to obtain the Hessian in the form of

$$\nabla_w \nabla_w \log p(\boldsymbol{w}|\boldsymbol{y},\boldsymbol{\alpha})|_{W_{MP}} = -(\phi^T\boldsymbol{B}\phi + \boldsymbol{A}), \quad (3.18)$$

where $\mathbf{B} = diag(\beta_1, \beta_2, \cdots, \beta_N)$ is a diagonal matrix with $\beta_n = \sigma\{f(\boldsymbol{x}_n;\boldsymbol{w}_{MP})\}[1-\sigma\{f(\boldsymbol{x}_n;\boldsymbol{w}_{MP})\}]$. $\boldsymbol{\Phi}$ is the design matrix with$\Phi_{nm} = K(\boldsymbol{x}_n, \boldsymbol{x}_{m-1})$ and $\Phi_{n1} = 1$. This result is then negated and inverted to give covariance Σ, shown as follows, for a Gaussian approximation to the posterior over weights centered at $\boldsymbol{w}_{MP}$.

$$\Sigma^{-} = (\phi^T\boldsymbol{B}\phi + \boldsymbol{A})^{-1} \quad (3.19)$$

Consequently, the classification problem is locally linearized around $\boldsymbol{w}_{MP}$ in an effective way with the following.

$$\boldsymbol{w}_{MP} = \Sigma\phi^T\boldsymbol{B}\hat{t} \quad (3.20)$$

$$\hat{\mathrm{t}} = \phi\boldsymbol{w}_{MP} + \boldsymbol{B}^{-1}(y-f) \quad (3.21)$$

These equations are fundamentally equivalent to the solution of a generalized least-squares problem. After obtaining $\boldsymbol{w}_{MP}$ the hyperparameters $\alpha_i$ are updated using $\alpha_i^{new} = \lambda_i / w_i^2$, where $w_i^2$ is the *i*th posterior mean weight and $\lambda_i$ is defined as $\lambda_i = 1 - \alpha_i \Sigma_{ii}$ where $\Sigma_{ii}$ is the *i*th diagonal element of the covariance and can be regarded as a measure of how well determined each parameter $w_i$ is by the data. During the optimization process, many $\alpha_i$ will have large values. Therefore, the corresponding model weights are pruned out, producing sparsity. The optimization process typically continues until the maximum change in $\alpha_i$ values is below a certain threshold or the maximum of iteration number of iterations is reached.

Using the "Sparse Logistic Regression ToolBox" software [21], I applied liner RVMs and nonlinear RVMs for two-vowel classification. RVM-L and RVM-G respectively use

$$\Phi(x) = \begin{cases} x & \text{RVM-L} \\ e^{-\frac{\|x - x_j\|^2}{2\sigma}} & \text{RVM-G} \end{cases} . \tag{3.22}$$

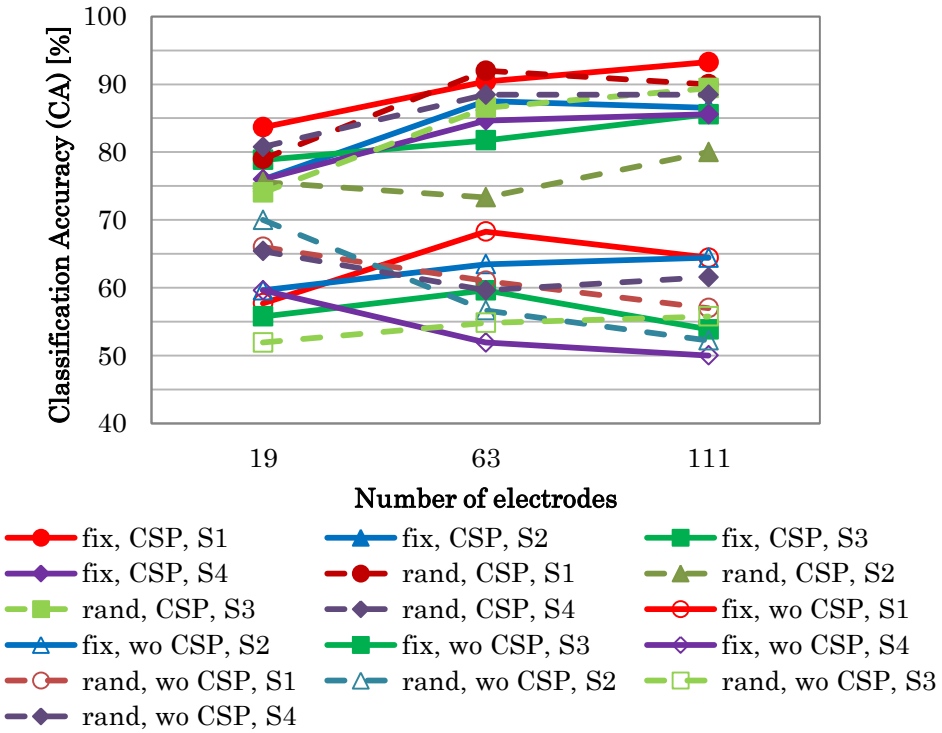
as $\Phi(x)$ in Eq. (3.15)

In Fig. 3.1, when classifier blocks act as RVM-L and RVM-G classifiers, the input is *x* and $x_j$ in Eq. (3.22) and the output is classification results.

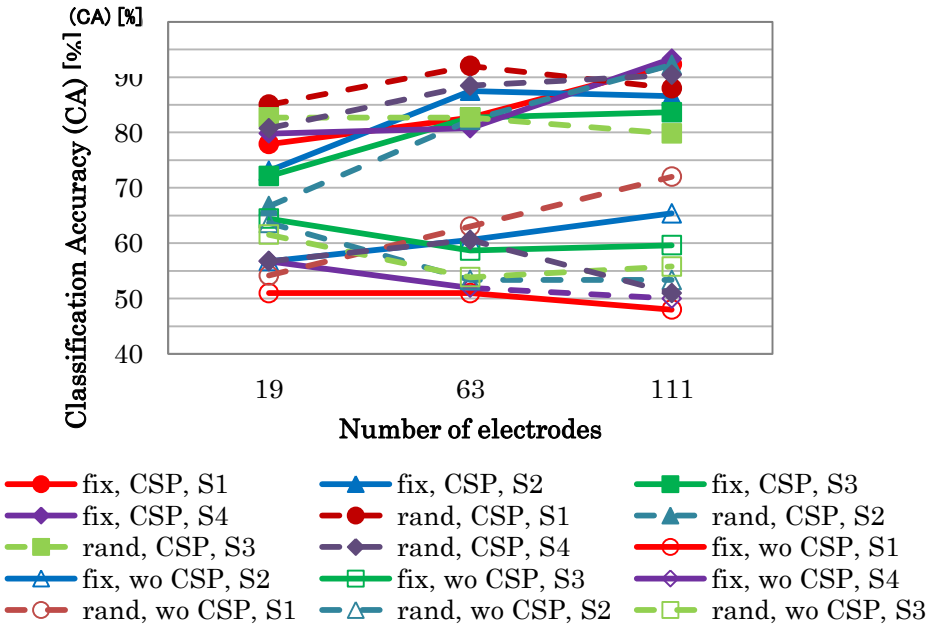
## 4. Results

### 4.1. Pairwise classification results for /a/ vs. /u/ and /u/ vs. /o/

Figs. 4.1(a) and (b) show the classification accuracy (CA) of two-vowel classifications, /a/ vs. /u/ and /u/ vs. /o/, when the SVM hyperparameter $C$ is set to 10 and parameter σ is optimized. In Fig. 4.1, the horizontal axis is the number of electrodes. The legend symbols “fix” and “rand” respectively denote the fixed order task and the random order task. “CSP” and “wo CSP” respectively denote with CSP and without CSP. “S1”, “S2”, “S3”, and “S4” respectively denote subject 1, subject 2, subject 3, and subject 4. In these calculations, the number of collection elements $M$ is set to 20. It shows the effect of CSP over the subjects, the tasks, and combinations of vowels. No significant difference was found between the fixed order task and the random order task. CAs with CSPs were better than those without CSP. The averaged improvement for 19 electrodes, 63 electrodes, and 111 electrodes were respectively 17%, 22%, 24%. The CAs using CSP with 19 electrodes are worse than those with 63 electrodes, and that with 63 electrodes were slightly worse than those with 111 electrodes except for “rand, CSP, S2”. In comparison to 111 electrodes, the degradation of the CA of 19 electrodes averaged over subjects, vowel combinations, and the tasks was about 9%.




(a) /a/ vs. /u/

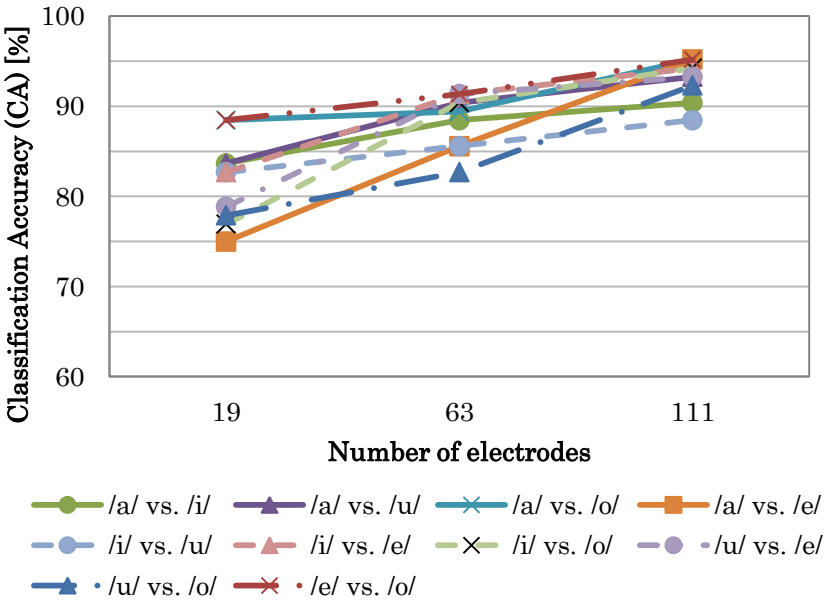



(b) /u/ vs. /o/

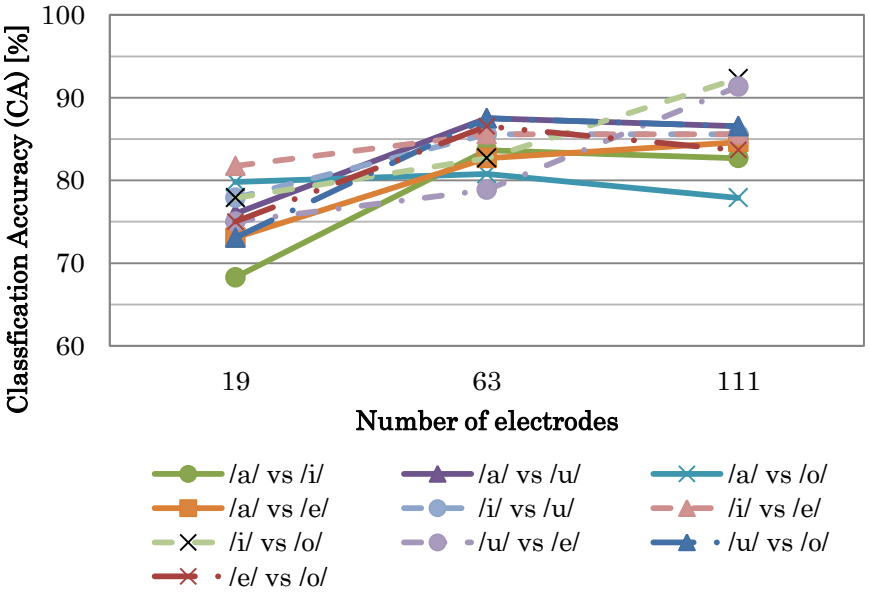
**Figure 4.1** Figure Classification results

### 4.2. Pairwise classification results of all vowel combinations

Figs. 4.2(a) and (b) show the CA of all two-vowel combinations for the five vowels. The other conditions and the meanings of the legend symbols were as shown in Fig. 4.1. In the 63 electrodes and 111 electrodes the CAs over the entire vowel combinations were better than 73%. Furthermore, the average CAs of 19 electrodes, 63 electrodes, and 111 electrodes were 78%, 85%, and 87%, respectively. In Fig. 4.2(b), I found that the CA for 111 electrodes was worse than that for 63 electrodes only in the case of /a/ vs. /i/ and /a/ vs. /o/ and S2. The cause was that one electrode's signal was too dominated using CSP in 111 electrode and the adaptive collection collected many elements of the same signal. As a result, it became easy to be influenced by fluctuation.




(a) S1 (S1, CSP, fix)




(b) S2 (S2, CSP, fix)

**Figure 4.2** Classification results for all two-vowel combinations.

**Table 4.1** Classification accuracy
Compared with the previous study
(/a/ vs. /u/, 63(64) electrodes, random order)

| My study | | | | Previous |
|---|---|---|---|---|
| **S1** | **S2** | **S3** | **S4** | **Study** [8] |
| **92%** | **73%** | **88%** | **87%** | **56–72%** |

**Table 4.2** Average classification accuracy
comparison of tasks
(63 electrodes, average of subjects)

| My study | |
|---|---|
| **FIX** | **RAND** |
| **86%** | **85%** |

### 4.3. Comparison with the previous study

Table 4.1 shows a comparison with results obtained in a previous study [8] in the condition of /a/ vs. /u/, the random order, 63 electrodes while the previous study used 64 electrodes. My method employing the adaptive collection achieved 73–92% of CA, whereas the previous study caused 56–72% of CA. That result demonstrates that my method was superior to that of the previous study.

### 4.4. Comparison of the tasks

Table 4.2 shows CAs averaged over vowel combinations and subjects in the condition of fixed order task and random order task with 63 electrodes. Those results show slightly different between the tasks. It means that there is no significant difference between fixed order task and random order task.

### 4.5. Classification results obtained using RVM-L and RVM-G

Fig. 4.3 shows the averaged classification accuracies (CAs) over all pairwise classifications for each subject using RVM-L and RVM-G in the case of fixed order tasks and use of CSP and 19channel brainwaves. For the RVM-G, hyperparameter $\sigma$ is optimized. The legend symbols "S1"–"S5" respectively denote data for subjects 1–5. In these calculations, the number of collection elements $M$ is set to 20.

The CAs using RVM-L are worse than those using RVM-G. CAs using RVM-L are around the chance level of 50%. However, CAs using RVM-G are 75%–87%. Taken together, these results indicate that linear classification is ineffective for silent speech of which features are nonlinear.

### 4.6. Classification results using SVM–G and RVM-G

Figs 4.4(a) and (b) show the averaged classification accuracies (CAs) over all pairwise classifications for each subject using SVM-G and RVM-G. For SVM-G and RVM-G, hyperparameter $\sigma$ is optimized. For SVM-G the hyperparameter $C$ is set to 10 to reduce the calculation cost. The legend symbols "CSP" and "wo CSP" respectively denote results obtained with CSP and without CSP. Other conditions and legend symbols are as shown in Fig. 4.3. Fig. 4.4(a) is for fixed order tasks. Fig. 4.4(b) is for random order tasks.

These results show the effect of CSP over all subjects, all tasks, and all combinations of vowels. No significant difference was found between the fixed order task and the random order task. For the fixed order task, the performance of RVM-G is slightly better than SVM-G for S1, S2, and S5, however for other two subjects the performance of RVM-G was poorer than SVM-G. In contrast, for random order tasks, the performance of RVM-G is slightly better than SVM-G for S1 and S4, however for other three subjects the performance of RVM-G was poorer than SVM-G. Considering the standard deviation in 4–9%, no significant difference was found between SVM-G and RVM-G.

Results show that averaged CAs using SVM-G and RVM-G are, respectively, 77% and 79%.

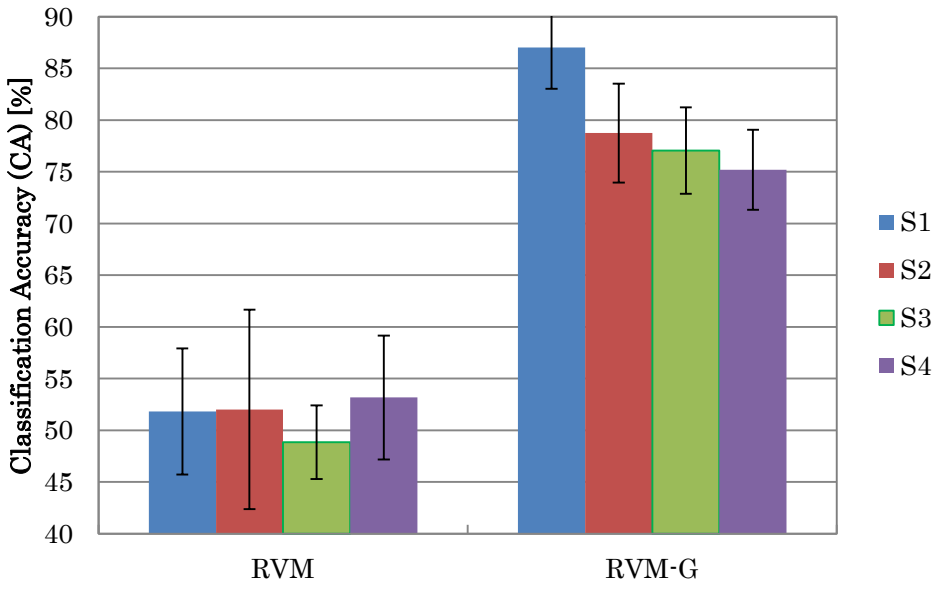
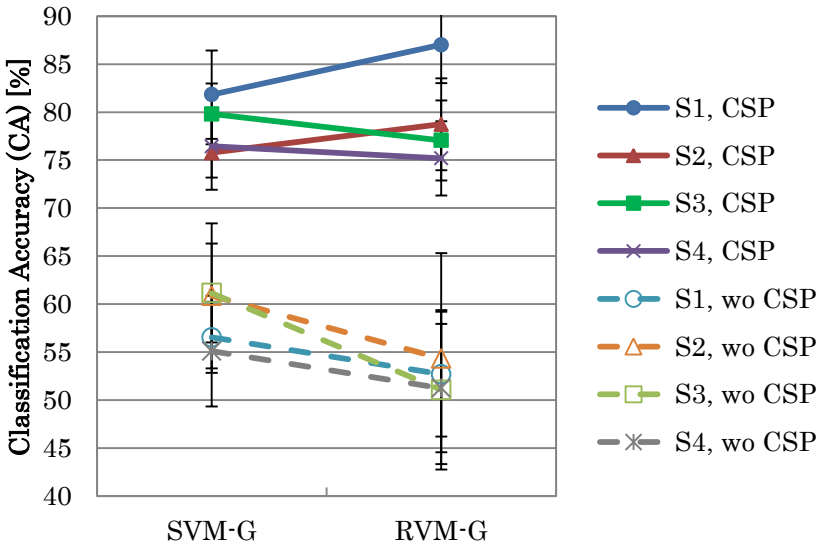
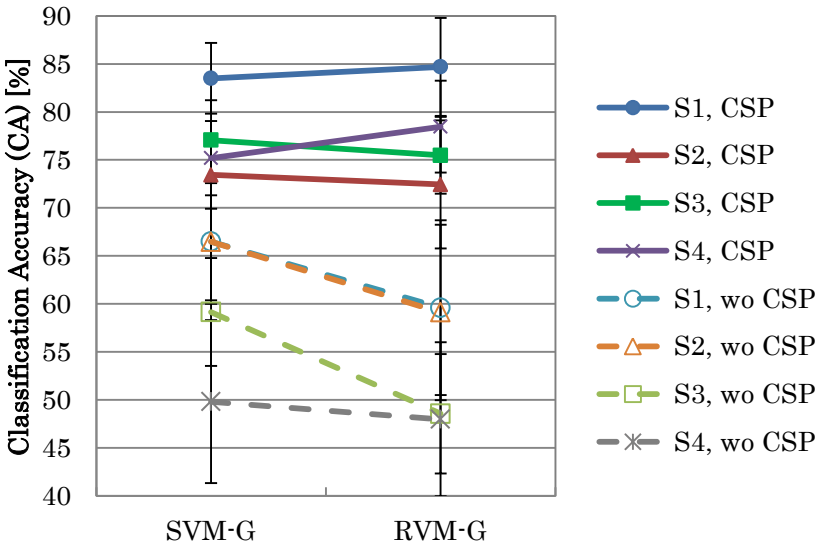



**Figure 4.3** Classification results (RVM-L and RVM-G).




(a) Fixed order task




(b) Random order task

**Figure 4.4** Classification results (SVM-G and RVM-G).

### 4.7. Features of effective vectors

Table 4.3 presents percentages of efficient vectors normalized by the number of training data, in the case of using CSP, 19 channel brainwaves, fixed order tasks, and random order tasks.

No significant difference was found between the fixed order task and the random order task. Based on these results, SVM-G used about 97% of the training data as vectors. RVM-G used about 55% of training data. RVM-L used less than 9%, which is too small because CAs using RVM-L were approximately chance level. The CAs of RVM-G were approximately equal to that of SVM-G, even though the vectors were fewer.

**Table 4.3** Percentages of effective vectors

| | ***SVM-G*** | ***RVM-G*** | ***RVM-L*** |
|---|---|---|---|
| ***FIX*** | 97% | 56% | 8.1% |
| ***RANDOM*** | 97% | 53% | 8.7% |

## 5. Discussion

For this study, I attempted to use classification algorithms for speech prostheses with an imagined voice vocalization, silent speech. Some major conclusions derived from the results are the following.

First, in the 63 electrodes and 111 electrodes, the CAs over the entire vowel combinations were better than 73% (Fig. 4.2). Furthermore, the average CAs of 63 electrodes in the condition of order task and random order task were respectively 86% and 85% (Table 4.2). Fagan et al. [6] achieved accuracy of 94% for phoneme detection using magnet implantation around a patient's mouth. Results show that my method demonstrated near performance without the surgical operation. However, a feasibility problem exists in relation to 63 electrode application. In my method, the average CA with 19 electrode application was only 78%. The reduction of number of electrodes remains as a subject for future study.

Second, in this study, 73%–92% of the CA was attained with the use of the adaptive collection for 63 electrode measurement and /a/ vs. /u/. It showed better performance than earlier research, in which the CAs were 56%–72% in nearly identical conditions [8]. That improvement results from the adaptive collection. The adaptive collection selects elements using the evaluation data for classification. The evaluation data and the test data are of different epochs measured during the same experiment. In other words, the adaptive collection depends on stability of the brain signals which obtained during an experiment. Because the adaptive collection uses trend of elements, the signal feature during the experiment must be stable. Oppositely, I confirmed that different elements were collected in the case of data measured in the other date. These results appear to be contradictory. One possibility is that the spatial feature of brain waves changes over a long duration, but it remains the same during the short duration because the change is slow or it does not occur in continuous trials. I shall examine this point further in future studies.

Third, in comparison to 111 electrodes, the degradation of the CA of 19 electrodes averaged over subjects, vowel combinations, and the tasks was about 9%, although Yong et al. [24] showed that the CA reduction from 118 electrodes to 13 electrodes was only 3.8% in the case of classification between the right hand and right foot. The difference depends on the objects. This study reduced fixed electrodes for the feasibility study and Yong et al. used sparseness. The difference is also attributable to the difference of the classifying objects. Whereas Yong et al. classified data obtained with

the right hand and right foot in the imagined motor task, I classified the vowels in the silent speech task. According to the geometries of the motor cortex [25], significant distance exists between the motor areas of the hand and foot. However, the vowel classification must classify the actions of the tongue, lip, glottidis, and so on. It needs more resolution than the hand or foot detection. 7% out of the 9% was the effect of CSP.

Fourth, results show that each subject had different suitable elements for the vowel classification as shown in the previous study [26]. I also confirmed that different suitable elements were related to the vowel combinations, and dates.

Fifth, the relevance vector machine (RVM) was proposed as a method providing fewer relevance vectors than support vector machine (SVM) [27]. Results show that using RVM-G instead of SVM-G reduced the ratio of the number of efficient vectors to the number of training data from 97% to 55% (Table 4.3). At this time, the averaged classification accuracies (CAs) using SVM-G and RVM-G were, respectively, 77% and 79% (Figs. 4.4(a) and (b)). That is, results show that RVM-G reduces the number of vectors without degradation of CAs. In this case, the condition was using 19 electrodes with CSP filter and AC, and the number of training data was 51.

Sixth, results show that CAs using RVM-G were weaker than SVM-G when the training data were few. Fig. 6.1(a) shows the relation between CAs and the number of training data, and the relation between the number of vectors and the number of training data. The CAs of RVM-G are worse than those of SVM-G when the training data are few, even when the quantities of vectors are nearly equivalent. This point is important because reduction of training data is proposed as a method for online processing [28]. RVM might therefore be unsuitable for such method.

Seventh, RVM entails huge calculation costs for optimization when the number of training data is large. When the number of training data is 51, the calculation costs of RVM-L and RVM-G normalized by that of SVM-G were about 1.5 and 4, respectively. It means that the effect of the calculation cost for optimization is larger than that for the number of vectors. Fig. 6.1(c) shows the relation between the normalized calculation cost and the number of training data. The normalized calculation cost is operating time for RVM-G dived by that for SVM-G. When the number of training data increases, the calculation cost increases in proportion to triplicate ratio [27]. However, when the number of training data is few, CAs using RVM-G is weak (Fig. 6.1(a)). The trade-off is difficult problem. For implementation for online processing, one must choose carefully which part functions as serial processing.

Eighth, results show that a linear classifier does not work for classification of silent speech, i.e., RVM-L was ineffective (Fig. 4.3). Some linear classifiers, sparese logistic regression (SLR) and regularized logistic regression (RLR), which I tried to use were also ineffective for the classification of silent speech. However, Gaussian kernel entails the problem that hyperparameter σ must be optimized by cross validation, of which the calculation cost is high. Finding some method for automatic optimization or another appropriate kernel, which needs no cross validation, remain as issues for future study.

Ninth, the greater the number of collected elements *M* of AC, the higher the calculation cost. Through this study, *M* was set to 20 using a heuristic approach. It is needs to study the reduction method without degradation of performance.

In this study, I evaluated pairwise classifications and did batch processing. Multiple classifications and online processing remain as subjects for future study.

## 6. Conclusion

The overall aim of this study was to show the feasibility of speech assistive BCI using silent speech.

I published two papers [9, 10]. The first paper showed results as below:

Previous study found a 56–72% of CA for /a/ vs. /u/ using 64 electrodes, however, using adaptive collection, it were improved to 73–92%. The degradation of CA using 19 electrodes from 111 electrodes was about 9%. Using 63 electrodes, more than 73% of CA was achieved for all combinations of the five vowels and the average was 85%.

The next paper showed results as below:

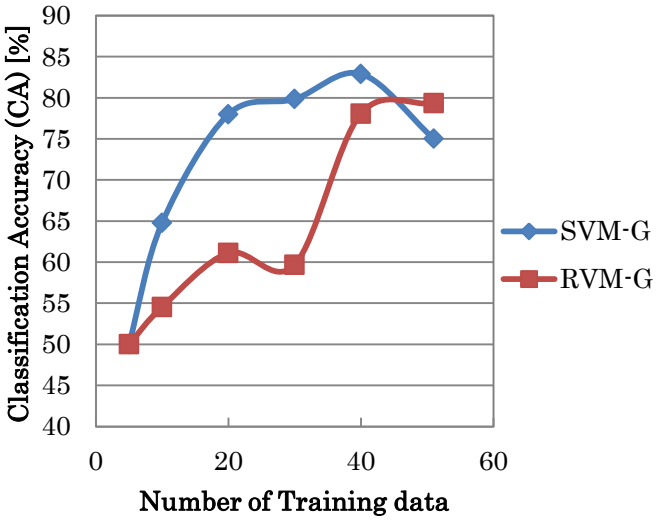
No significant difference was found between RVM-G and SVM-G of CAs. Using RVM-G instead of SVM-G reduced the ratio of efficient vectors to training data from 97% to 55%. When the number of training data was 51, the calculation costs of RVM-G was 4 times of SVM-G because of optimization. CAs using RVM-G were weaker than SVM-G when the number of training data were few. Linear classifier did not work for classification of silent speech. As a result, RVM is unsuitable for silent speech classification.

From these results, conclusions are:

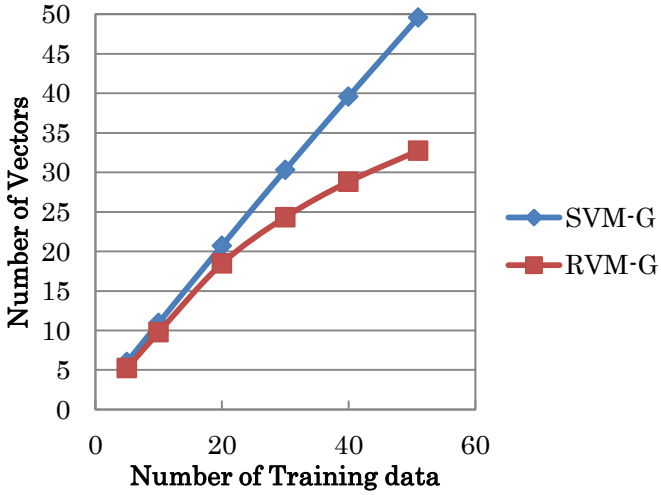
The adaptive collection, which I proposed exhibited great potential for use in classification of imagined voice for a speech prosthesis controller. The best classification method for silent speech is nonlinear SVM so far. The feasibility of speech assistive BCI using silent speech needs more study in the future.

**Acknowledgement**

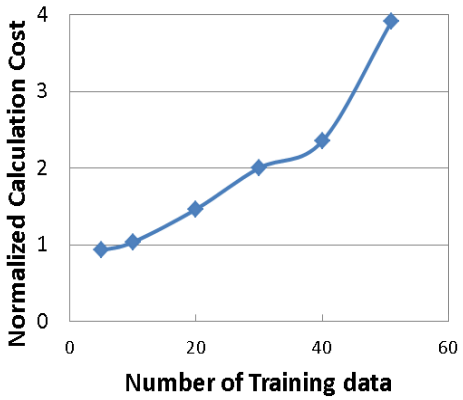
I would like to acknowledge cooperative members who helped me as subjects and experiment's assistants in Professor Dr. Hori's laboratory, Dr. Abe's laboratory, Dr. Hayashi's laboratory and etc. of the graduate school of science and technology, Niigata University. I thank Professor emeritus Dr. Miyakawa and Professor Dr. Hori for their accommodating attitude.




(a) Classification accuracies




(b) Number of vectors




(c) Calculation cost

**Figure 6.1** Effect of training data quantity (SVM-G and RVM-G).

**References**


[1] E. Donchin, K. Spencer, and R. Wijesinghe, “The mental prosthesis: assessing the speed of a P300-based brain–computer interface,” IEEE Trans. Rehabil. Eng. Vol.8, no.2, pp.174–179, 2000.

[2] HJ Hwang, JH Lim, YJ Jung, H Choi, SW Lee, “Development of an SSVEP-based BCI spelling system adopting a QWERTY-style LED keyboard,” Journal of Neuroscience, Elsevier , Volume 208, Issue vol.1, no.6, pp.59–65, 2012.

[3] J.L. Trejo, R. Rosipal, and B. Matthews, “Brain–computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials,” IEEE Trans. Neural Systems Rehabil. Eng., vol.14, no.2, pp.225–229, 2006.

[4] M. Naito, Y. Michika, K. Izawa, Y. Ito, M. Kiguchi, and T. Kanazawa, “A communication means for completely locked-in ALS patients based on changes in cerebral blood volume measured using near-infrared light,” IEICE Trans., Inf. Syst. E90-D, no.7, pp.1028–1037, 2007.

[5] M. Wand and T. Schultz, “Towards speaker-adaptive speech recognition based on surface electromyography,” In: Internat. Conf. on Bioinspired Systems and Signal Processing, Presented at the Biosignals 2009, Porto, Portugal, 2009.

[6] M. Fagan, S. Ell, J. Gilbert, E. Sarrazin, and P. Chapman, “Development of a (silent) speech recognition system for patients following laryngectomy,” Med. Eng. Phys. vol.30, no.4, pp.419–425, 2008.

[7] H.F. Guenther, S.J. Blumberg, J.E. Wright, A. Nieto-Castanon, A.J. Tourville, M. Panko, et al. “A wireless brain-machine interface for real-time speech synthesis,” PLoS ONE, vol.4, no.12, e8218, 2009.

[8] C.S. Dasalla, H. Kabara, M. Sato, and Y. Koike, “Single-trial classification of vowel speech imagery using common spatial patterns,” Neural Networks vol. 22, no.9, pp.1334–1339, 2009.

[9] M. Matsumoto and J. Hori, “Classification of Silent Speech using Adaptive Collection,” Computational Intelligence in Rehabilitation and Assistive Technologies (CIRAT), 2013 IEEE Symposium on, April 2013.

[10] M. Matsumoto, J. Hori, “Classification of Silent Speech using Support Vector Machine and Relevance Vector Machine,” Applied Soft Computing, 2013.11

[11] R.C. Oldfield, “The assessment and analysis of handedness: The Edinburgh inventory,” Neuropsychologia, vol.9, pp.97–113, 1971.

[12] EASY CAP 128-Channel-Arrangement, http://www.easycap.de/easycap/e/electrodes/11_M15.htm.

[13] http://www.biosemi.com/headcap.htm.

[14] 10–20 system wikipedia,

[15] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, “Designing optimal spatial filters for single-trial EEG classification in a movement task,” Clinical Neurophysiology 100, pp.787–798, 1999.

[16] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller, “Optimal spatial filtering of single trial EEG during imagined hand movement,” IEEE Transactions on Rehabilitation Engineering, vol.8, no.4, pp.441– 446, 2000.

[17] A. Rakotomamonjy, “Variable selection using SVM based criteria,” The Journal of Machine Learning Research Archive, vol.3, pp.1357–1370, 2003.

[18] E.B. Boser, M.I. Guyon, and N.V. Vapnik, “A training algorithm for optimal margin classifiers,” In: D. Haussler (Ed.), Fifth Annual ACM Workshop on COLT, ACM Press Pittsburgh, PA, pp.144–152, 1992.

[19] F. Asano, M. Kimura, T. Sekiguchi, and Y. Kamitani, “Classification of movement-related single-trial MEG data using adaptive spatial filter,” IEEE ICASSP, pp.357–360, 2009.

[20] C. Hsu, C. Chang, and C. Lin, “A practical guide to support vector classification,” Retrieved from, 2003.
http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[21] O. Yamashita, “Sparse Logistic Regression ToolBox,” 2009.

[22] M.E. Tipping, “Sparse Bayesian learning and the Relevance Vector machine,” Journal of Machine Learning Research 1, pp.211–244, 2001.

[23] M.E. Tipping, “The Relevance Vector Machine,” In: S.A. Solla, T.K. Leen, and K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, MIT Press pp.652-658, 2000.

[24] X. Yong, R.K. Ward, and G.E. Birch, “Sparse Spatial Filter Optimization for EEG Channel Reduction In Brain-Computer Interface,” Acoustics, Speech and Signal Processing, (ICASSP) 2008. IEEE International Conference on pp.417–420, 2008.

[25] Wikipedia Cortical homunculus,
http://en.wikipedia.org/wiki/Cortical_homunculus.

[26] E.C. Leuthardt, C. Gaona, M. Sharma, N. Szrama, J. Roland, Z. Freudenberg, J. Solis, J. Breshears, and G. Schalk, “Using the electrocorticographic speech network to control a brain-computer interface in humans,” J. Neural. Eng. In press, 2011.

[27] C.M. Bishop, “Pattern Recognition and Machine Learning,” Springer, pp.345–356, 2006.

[28] T. Asano, H. Nakayama, and T. Tanino, “Incremental learning and forgetting using Sensitivity in AVM,” IEICE, vol.104, no.760, pp.171–176, 2005.