PAPER

Inferring imagined speech using EEG signals: a new approach using Riemannian manifold features

To cite this article: Chuong H Nguyen *et al* 2018 *J. Neural Eng.* **15** 016002

IOP Publishing Journal of Neural Engineering

J. Neural Eng. **15** (2018) 016002 (16pp) https://doi.org/10.1088/1741-2552/aa8235

Inferring imagined speech using EEG signals: a new approach using Riemannian manifold features

Chuong H Nguyen, George K Karavas and Panagiotis Artemiadis[1]

School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ 85287, United States of America

E-mail: chuong.h.nguyen@asu.edu, georgioskonstantinos.karavas@asu.edu and panagiotis.artemiadis@asu.edu

Received 25 March 2017, revised 22 July 2017
Accepted for publication 26 July 2017
Published 1 December 2017

Abstract

Objective. In this paper, we investigate the suitability of imagined speech for brain–computer interface (BCI) applications. *Approach*. A novel method based on covariance matrix descriptors, which lie in Riemannian manifold, and the relevance vector machines classifier is proposed. The method is applied on electroencephalographic (EEG) signals and tested in multiple subjects. *Main results.* The method is shown to outperform other approaches in the field with respect to accuracy and robustness. The algorithm is validated on various categories of speech, such as imagined pronunciation of vowels, short words and long words. The classification accuracy of our methodology is in all cases significantly above chance level, reaching a maximum of 70% for cases where we classify three words and 95% for cases of two words. *Significance.* The results reveal certain aspects that may affect the success of speech imagery classification from EEG signals, such as sound, meaning and word complexity. This can potentially extend the capability of utilizing speech imagery in future BCI applications. The dataset of speech imagery collected from total 15 subjects is also published.

Keywords: EEG, BCI, speech imagery, relevance vector machines

(Some figures may appear in colour only in the online journal)

1. Introduction

In an effort to improve the quality of clinical neurology and rehabilitation, the aspiration of deciphering brain electrical signals to allow humans to directly interact with their environment has been an attractive research for the past decades. Originally developed to facilitate people with severe motor disabilities, the majority of research in brain–computer interface (BCI) applications has been devoted to interpret and discriminate motor imagery to control external devices, such as a wheelchair [1]. One fundamental drawback of using motor imagery as the control command is the limitation in the degrees of freedom (DoF). Most applications related to motor imagery rely on binary classification, such as classification of left versus right hand imagery, whereas the highest number of DoF is provided by four class classifications, e.g. left versus right hand, tongue and foot motion imagery. Another popular approach is to use visual evoked potentials. These are brain response signals that have the same frequency with a flickering stimulus, such as a light blinking with a specific frequency. In this way, we can create a 'visual touchscreen', i.e. screen on which the user will focus on a desired blinking spot that is associated to a specific command, such as the application in [2]. This approach however requires a considerable attention time to the active stimuli which are also constrained by color, frequency band and spatial resolution [3].

In an attempt to push BCI application out of the rehabilitation scope and make it useful for healthy individuals as well, the mentioned approaches become inadequate as it is much easier and accurate to use someone's hands with a joystick or a touchscreen. In other words, an efficient BCI for normal situations should allow the user to use his hands and vision freely to interact with other devices, and provide additional functionalities and DoF for a more complex task. This demand

[1] Author to whom any correspondence should be addressed.

inspires our research in exploring the applicability of speech imagery as an alternative control signal for BCI. Moreover, silent communication is also desired in many circumstances where sound recognition is prohibited, such as in a noisy corrupted surrounding or subjects with hearing or speaking disabilities. In this work, we use electroencephalography (EEG) signals in order to extract the necessary features because it is noninvasive and its application does not require specific medical knowledge.

This paper is organized as follows. Section 2 summarizes the previous works and our research objectives and contribution. The proposed approach and theoretical background are presented in section 3. Section 4 describes the experimental procedure, and section 5 discusses the data analysis and the main results. Section 6 compares the performance of our methods with several representative works found in the literature, and provides further discussion on the possible impacts to the speech imagery paradigm. Section 7 concludes the paper and discusses future work.

2. Related work and research objective

In recent years, several efforts of extracting human abstract thinking have been investigated. For example, Esfahani and Sundararajan [4] attempt to classify primitive shapes, such as cube, sphere, cylinder, pyramid and cone, imagined by users. The reported results are encouraging, as the average accuracy across ten subjects is 30.6−37.6% compared to 20% accuracy by chance. In their report, the highest accuracy is actually 54.1%. In their following work [5], Contreras and Sundararajan are able to discriminate three shapes, e.g. cone, rectangle and cylinder, with 60% accuracy using the shapelet feature. Although these results are promising, performing visual imagery requires high concentration and is hard to maintain good performance throughout the task, as the authors were unable to find a repeated shapelet's pattern.

In contrast to visual imagery and motor imagery, speech imagery is quite consistent between users and easier to perform and repeat without the requirement of initial training. In [6], Herff and Schultz conducted a review of techniques deciphering neural signals for automatic speech recognition. These techniques can be classified based on the methods of collecting neural signals, which include metabolic signals or electrophysiological signals. Functional magnetic resonance imaging (fMRI) and functional near infrared spectroscopy (fNIRS) are two main approaches to collect metabolic signals. However, they are limited to clinical environment only. To record electrophysiological signals, microarray, EEG, magnetoencephalography and electrocorticography are used. Among them, EEG is more suitable to daily activities as it is noninvasive and easy to setup, while the others either require a certain level of clinical treatments or are too bulky. Besides neural signals, silent speech can also be recognized based on electromyographic (EMG) activity in facial muscles, such as in the works conducted by Denby *et al* [7] and Schultz [8].

In this article, we are interested in deciphering imagined speech from EEG signals, as it can be combined with other mental tasks, such as motor imagery, visual imagery or speech recognition, to enhance the degree of freedom for EEG-based BCI applications. Furthermore, it can also be applied to locked-in patients where recording facial EMG activities is prohibited [9]. Interpreting imagined speech from EEG signals however is challenging and still an open problem.

Current research has investigated two possibilities, using phonemes or syllables, in which a user imagines saying vowels, e.g. /u/ or /a/, or words, e.g. left or right, without overt vocalization. In [10], Wester *et al* created a system apparently capable of recognizing imagined speech with high accuracy rate. However, Porbadnigk *et al* [9] later revealed that the successful recognition accuracy in [10] is mainly due to the experiment process of collecting data, which accidentally created temporal correlation on EEG signals. That is, if the words are collected in a block, i.e. each word is repeated 20 times before trying another word, the recognition rate is high. However, if the words are recorded in a random order, the accuracy dropped to the chance level.

D'Zmura *et al* [11] conducted a speech imagery experiment with four participants. Wavelet envelops in theta (3–8 Hz), alpha (8–13 Hz) and beta (13–18 Hz) bands were extracted as features by using Hilbert transform. The highest accuracy is reported in the range 62–87% with discriminant features in beta band. This is in contrast to the report of Kim *et al* [12], where alpha band is discovered as the most discriminant band based on ANOVA test and common spatial patterns (CSPs). In the work conducted by Brigham and Kumar [13], 7 participants performed speech imagery with two syllables, /ba/ and /ku/. By using Hurst score, only 8–15 trials from total 120 trials were selected as meaningful samples for classification. The lowest and highest accuracy is reported as 46% and 88% using 13 and 11 trials, respectively. The result however is not statistically persuasive. In [14, 15], DaSalla *et al* obtained 68–79% accuracy when classifying three stages /a/, /u/, and rest, by using CSP. However, the results of CSP point out that the discriminant channels are Fz, C3, Cz, and C4. This in turn suggests that the results are mainly due to the activation of motor cotex rather than speech imagery. In a similar experiment, Deng *et al* [16] used Huang–Hilbert transform (HHT) to perform adaptive frequency transform, as HHT is more suitable for non-stationary, nonlinear signal like EEG than Fourier transform or wavelet transform. Using the spectral power density in the frequency band 3–20 Hz as feature, the highest accuracy rate is reported at 72.6% of classifying /ba/ and /ku/ vowels. Idrees and Farooq [17] proposed an approach, in which 11 features are extracted, such as mean, variance, skewness, kurtosis, geometric mean, harmonic mean, inter-quartile range, energy sum, entropy, standard deviation and waveform length, and linear discriminant analysis (LDA) classifier is used on those features. Testing on the dataset published by DaSalla *et al* [14], the author obtained the classification accuracy from 65–82.5% which is approximately 2–17.5% higher than that from DaSalla *et al* [14]. Other features, such as Gabor filter [18] and Mel frequency cepstral coefficients [19] have been also proposed and obtained encouraging results.

In terms of classifying speech imagery of words, the literature is more limited. For example, in [20] Suppes *et al* performed an experiment with seven subjects in which they were

able to distinguish between seven words during an auditory comprehension task using EEG signals. Five of those subjects were also performing speech imagery during the task. Later, in [21], the authors were able to apply their methods on sentences heard by the subjects. In [22], Gonzalez-Castaneda *et al* were able to classify between five different imagined words, namely 'up', 'down', 'left', 'right' and 'select'. Utilizing appropriate frequency transform techniques, such as discrete wavelet transform or Bump model combined with the bag-of-words algorithm, the EEG signal can be treated as audio signal (sonification) or sentences of words in a text (textification). Their analysis included 27 subjects and they were able to report classification accuracies of 83.34% on average when textification techniques were used. In [23], Salama *et al* were able to distinguish between two imagined words, namely 'Yes' and 'No', using different types of classifiers such as support vector machines (SVMs), discriminant analysis, self-organizing map, feed-forward back-propagation and by combining multiple classifiers. Their analysis produced average accuracies between 57% and 59% among seven subjects. In [24], Wang *et al* were able to distinguish between two Chinese characters that meant 'left' and 'one', respectively, and the rest state by using CSPs and SVMs. Their highest accuracies were between 73.65% and 95.76% when comparing between each of the imagined words and the rest state. However, the accuracies of classification between the two words themselves was lower, e.g. 82.3% mean value. The interesting idea about their work is that they used two different electrode montages; one covered the whole brain area and included 30 channels, and another with 15 channels that covered Broca's area and Wernicke's area which lie on the left brain hemisphere. No specific effect on the classification accuracy was reported from the choice of setup though. Finally, in [25], Mohanchandra *et al* were able to classify between five different words, namely 'water', 'help', 'thanks', 'food' and 'stop', by combining a one-against-all multiclass SVM with the subset selection method which was based on a set of principal representative features in order to reduce the dimensionality of the EEG data. They achieved average accuracies that ranged from 60% to 92% for the recognized words.

The aforementioned works have indicated that speech imagery is a promising approach, and the accuracy can be improved by selecting appropriate features and classifiers. The reported results however are often applied to pairwise classification, while the simultaneous recognition of multiple states can be more challenging. Furthermore, the lack of public datasets makes it hard to replicate the results as well as develop and compare different algorithms. This analysis hence motivates our research which leads to the following contributions:

- We propose a novel method to classify speech imagery. The proposed method can perform multi-class recognition simultaneously, and significantly outperforms other methods found in the literature in terms of accuracy and sensitivity.
- We investigate what are the important factors affecting the performance of classifying speech imagery. For example, if different words can be recognized accurately, is it because of the difference in their complexity, meaning or their sound?
- Our experimental data are analyzed to verify that they indeed correspond to speech imagery tasks, that can be discriminated with significant accuracy. The dataset is published[2] to assist community further research in the field.

3. Proposed method

Our proposed method can be summarized in four main steps. First, we extract low level features from the preprocessed EEG signal. These features are extracted at each time instant and are considered as local features. Second, a covariance matrix descriptor is used to fuse them together in order to further boost their global discriminative power for the entire imagination period. Since the covariance matrix lies in the Riemannian manifold, appropriate metrics need to be used to discriminate them. Hence, in the third step, we extract their tangent vectors as the final high level features vector, which are consequently fed to a relevance vector machine (RVM) to classify their labels. The approach is described in more detail below.

3.1. Theoretical background

3.1.1. Notation. Let $\mathbb{R}^n$ be an n dimension real space, $\mathbf{1}_n \in \mathbb{R}^n$ be a vector with all entries equal to 1, and $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is the identity matrix. $E\{\boldsymbol{x}\}$ is the expectation of $\boldsymbol{x}$ and $\mathrm{diag}(\boldsymbol{x})$ is the diagonal matrix constructed from $\boldsymbol{x}$. $\boldsymbol{X}(i,:), \boldsymbol{X}(:,j)$ and $X(i,j)$ denote the row vector i, the column vector j and the entry (i,j) of matrix $\boldsymbol{X}$, respectively. $\boldsymbol{A}^{\mathrm{T}}$ is the (conjugate) transpose of $\boldsymbol{A}$, and $\mathrm{vec}(\boldsymbol{A})$ be the vectorizing operator on matrix $\boldsymbol{A}$. If $\boldsymbol{A}$ is symmetric then $\mathrm{vec}(\boldsymbol{A})$ only takes an upper half of the matrix. Furthermore, we denote $\|\cdot\|_0$ as the $\mathcal{L}_0$ (i.e number of non-zero elements), and $\|\cdot\|$ as either $\mathcal{L}_2$ or Frobenius norm for vector or matrix respectively.

Definition 3.1. $\boldsymbol{x}|\mu,\alpha \sim \mathcal{N}(\boldsymbol{x}|\mu,\alpha^{-1})$ denotes that the random variable $\boldsymbol{x}$ follows a Gaussian distribution with mean μ and variance $\sigma^2 = \alpha^{-1}$, i.e, its probability $P(\boldsymbol{x}|\mu,\alpha) = \mathcal{N}(\boldsymbol{x}|\mu,\alpha^{-1})$.

Definition 3.2. An $n \times n$ matrix $\boldsymbol{A}$ is symmetric positive definite (SPD) if $\boldsymbol{A} = \boldsymbol{A}^{\mathrm{T}}$, $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{x} > 0, \forall \boldsymbol{x} \neq 0$. Equivalently, the eigenvalues of $\boldsymbol{A}$, denoted as $\lambda(\boldsymbol{A})$, are positive. An SPD matrix is considered as a point on Riemannian manifold denoted by $\mathrm{Sym}_{\mathrm{n}}^{+}$ [26].

Definition 3.3. Let $\mathcal{F}$ be the *Hilbert space* associated with an *inner product* $\langle\cdot,\cdot\rangle$. A *feature map* $\Phi : \mathcal{X} \mapsto \mathcal{F}$, where $\mathcal{X}$ is the *original space* and $\mathcal{F}$ is the *feature space*, defines a unique SPD kernel by $K(x,y) = \langle\Phi(x),\Phi(y)\rangle$. On the other hand, for a given SPD kernel and its corresponding reproducing kernel Hilbert space, there exists an associated feature map [27]. Hence, in machine learning, the *kernel trick* refers to constructing the kernel map K directly without explicit defining the feature mapping Φ.

[2] The database can be found at https://www.dropbox.com/s/01k9c75j0x3jfb9/Dataset.zip?dl=0.

Definition 3.4. $\boldsymbol{A}^k$, $\exp(\boldsymbol{A})$ and $\log(\boldsymbol{A})$ of an SPD matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ are defined through its eigenvalues $\boldsymbol{\Lambda}$ and eigenvectors $\boldsymbol{U}$ as [26]:

$$\begin{aligned}\boldsymbol{A}^k &\triangleq \boldsymbol{U}\mathrm{diag}([\lambda_1^k,\ldots,\lambda_n^k])\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Lambda}^k\boldsymbol{U}^{\mathrm{T}},\\ \exp(\boldsymbol{A}) &\triangleq \boldsymbol{U}\mathrm{diag}([e^{\lambda_1},\ldots,e^{\lambda_n}])\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}e^{\boldsymbol{\Lambda}}\boldsymbol{U}^{\mathrm{T}},\\ \log(\boldsymbol{A}) &\triangleq \boldsymbol{U}\mathrm{diag}([\log(\lambda_1),\ldots,\log(\lambda_n)])\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}\log(\boldsymbol{\Lambda})\boldsymbol{U}^{\mathrm{T}}.\end{aligned}$$

3.1.2. Covariance matrix descriptor. Let $\mathbb{R}^{m\times n}, \boldsymbol{x}_i \in \mathbb{R}^{m\times n}$, $\boldsymbol{x}_i \in \mathbb{R}^m$ be the set of m feature extracted from n samples in one trial with zero mean, i.e. $\bar{\boldsymbol{X}} = \boldsymbol{X} - \mathrm{mean}(\boldsymbol{X})$, COV matrix is constructed as

$$COV = \frac{1}{n-1}\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^{\mathrm{T}} \in \mathrm{Sym}_{\mathrm{m}}^{+},$$

3.1.3. Distance on Riemannian manifold. Since SPD matrices are in Riemannian manifold, Euclidean distance is not effective to discriminate them. In [28], a detailed description and a comparison of the performance between different metrics on $\mathrm{Sym}_{\mathrm{n}}^{+}$ are conducted in the context of BCI application. In this work, to measure the distance between two SPD matrices, we focus on using the distance between their tangent vectors. Although tangent vector flattens the manifold and may not fully preserve the nonlinear discrimination of the original space $\mathrm{Sym}_{\mathrm{m}}^{+}$, it is more computationally efficient and yields almost equivalent performance with other metrics, such as log-det divergence or Kullback–Leibler divergence [28]. The tangent vector distance is summarized as follow.

- *The tangent vector* $\boldsymbol{T}$ of a point $\boldsymbol{S}$ at the reference point $\boldsymbol{C}$ is defined as.

$$\boldsymbol{T} = \log_{\boldsymbol{C}}\boldsymbol{S} \triangleq \log(\boldsymbol{C}^{-\frac{1}{2}}\boldsymbol{S}\boldsymbol{C}^{-\frac{1}{2}}). \tag{1}$$

- The distance between two points $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ on the Riemannian manifold can be derived through the Euclidean distance between the tangent vectors as

$$d_{TS}^2(\boldsymbol{S}_1,\boldsymbol{S}_2)_{\boldsymbol{C}} \triangleq \|\boldsymbol{T}_1 - \boldsymbol{T}_2\|_F^2.$$

The reference point $\boldsymbol{C}$ can be simply selected as the identity matrix, or the geometric mean of all training data points. If the geometric mean is used, this process is called normalization. In this work, the geometric mean of the covariance matrices is obtained through the tangent vectors of the training set as described in algorithm 1.

Algorithm 1. Mean of covariance matrices using tangent vector.

Input: Training dataset $\{\boldsymbol{S}_i\}_{i=1}^N \in \mathrm{Sym}_{\mathrm{m}}^{+}$.
Output: Mean of $\boldsymbol{S}_i$.
Procedure:

1. *Map each point to its tangent vector* $\boldsymbol{T}_i = \log(\boldsymbol{S}_i)$.
2. *Find the Euclidean mean* $\bar{\boldsymbol{T}} = \frac{1}{N}\sum_{i=1}^N \boldsymbol{T}_i$.
3. *Map the mean tangent vector to* $\mathrm{Sym}_{\mathrm{m}}^{+}$ $\bar{\boldsymbol{S}} = \exp(\bar{\boldsymbol{T}})$.

3.1.4. Relevance vector machine. In this paper, we use the multi class RVM (mRVM)[3] proposed by [29, 30], which is summarized as follows.

Let $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$ of N be a training set of N observations, each sample $\boldsymbol{x}_i$ has m features $\{\boldsymbol{x}_i^{(j)} \in \mathcal{X}_j\}_{j=1}^m$ in its original space $\mathcal{X}_j$ and a corresponding label $l_j \in \{1,\ldots,C\}$, where $C > 1$ is the number of classes.

We aim to build a model consisting of a multi-class hyperplane $\boldsymbol{W} \in \mathbb{R}^{(N+1)\times C}$ and a multi-kernel weighted vector $\boldsymbol{\beta} \in \mathbb{R}^m$ written as:

$$\underbrace{\begin{bmatrix} y_1 \\ \vdots \\ y_C \end{bmatrix}}_{\boldsymbol{y}(\boldsymbol{x})\in\mathbb{R}^C} = \underbrace{\begin{bmatrix} w_{10} & w_{11} & \ldots & w_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ w_{C0} & w_{C1} & \ldots & w_{CN} \end{bmatrix}}_{\boldsymbol{W}^{\mathrm{T}}\in\mathbb{R}^{C\times(N+1)}} \underbrace{\begin{bmatrix} 1 & \ldots & 1 \\ k_1(\boldsymbol{x}_1,\boldsymbol{x}) & \ldots & k_m(\boldsymbol{x}_1,\boldsymbol{x}) \\ \vdots & \vdots & \vdots \\ k_1(\boldsymbol{x}_N,\boldsymbol{x}) & \ldots & k_m(\boldsymbol{x}_N,\boldsymbol{x}) \end{bmatrix}}_{\boldsymbol{K}(x)\in\mathbb{R}^{(N+1)\times m}} \underbrace{\begin{bmatrix} \beta_1 \\ \vdots \\ \beta_m \end{bmatrix}}_{\beta\in\mathbb{R}^m}$$

where each element $\boldsymbol{K}_{ij}(\boldsymbol{x})$ is the kernel function evaluated at the training sample $\boldsymbol{x}_i$ using the feature $\boldsymbol{x}_i^{(j)}$. $\boldsymbol{y}(\boldsymbol{x}) = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{x})\boldsymbol{\beta}$ is the response of the model to a data sample $\boldsymbol{x}$. For a training sample $\boldsymbol{x}_i \in \boldsymbol{X}$, the response is

$$\boldsymbol{y}(\boldsymbol{x}_i) = [y_1 \ldots y_c \ldots y_C]^{\mathrm{T}}, \qquad y_c = \begin{cases} 1 & \text{if } l(x_i) = c, \\ 0 & \text{otherwise.} \end{cases}$$

and for a new sample $\boldsymbol{x}$, its label can be predicted as

$$l(\boldsymbol{x}) = c, \text{if } \boldsymbol{y}_c(\boldsymbol{x}) > \boldsymbol{y}_j(\boldsymbol{x})\ \forall j \neq c. \tag{2}$$

mRVM finds the optimal parameter $\boldsymbol{W}$ and $\boldsymbol{\beta}$ using the Bayesian rule with the following probabilistic constraints.

First, the true label $t_i = l(x_i)$ is assumed to be the measure of the prediction $y(x)$ corrupted by a standardized normal noise $\epsilon \sim \mathcal{N}(0,1)$, i.e $l(\boldsymbol{x}) = y(\boldsymbol{x}) + \epsilon$, or

$$P(t_i = c|\boldsymbol{x}_i, \boldsymbol{w}_c^{\mathrm{T}}, \beta) = \mathcal{N}(\boldsymbol{w}_c^{\mathrm{T}}\boldsymbol{K}(\boldsymbol{x}_i)\beta, 1). \tag{3}$$

Second, only a few samples in the training set are representative for its class, while the rest is redundant and safely ignored. This casts the sparsity on $\boldsymbol{W}$, which automatically solves the model's over-fitting problem and rejecting outliers. Hence, the model sparseness can be represented as

$$P(\boldsymbol{W}|\boldsymbol{\alpha}) = \mathcal{N}(\boldsymbol{W}|0, \boldsymbol{\alpha}^{-1}). \tag{4}$$

Third, since some features are more important than the other, we can set $\sum_{i=1}^m \beta_i = 1$, $\beta_i > 0$, which implies a Dirichlet distribution of $\boldsymbol{\beta}$, i.e.

$$P(\boldsymbol{\beta}|\boldsymbol{\rho}) = \mathrm{Dir}(\boldsymbol{\beta}|\rho_j). \tag{5}$$

Hence, the maximal magnitudes of $\boldsymbol{W}$ and $\boldsymbol{\beta}$ are controlled by $\boldsymbol{\alpha}$ and $\boldsymbol{\rho}$, which are also enforced to follow Gamma distribution, i.e.

$$P(\alpha_{ci}|\tau_{ci}, \upsilon_{ci}) = \gamma(\alpha_{ci}|\tau_{ci}, \upsilon_{ci}), \quad P(\boldsymbol{\rho}|\mu,\lambda) = \gamma(\boldsymbol{\rho}|\mu,\lambda). \tag{6}$$

The parameter $\Xi = [\tau, \upsilon, \mu, \lambda]$ can be automatically tuned as the arguments maximizing the evidence approximation, which is the following marginal likelihood function

[3] The Matlab code implementation of mRVM is published by the author Psorakis at https://github.com/ipsorakis/mRVMs.

$$P(l(\boldsymbol{X})|\boldsymbol{X},\Xi) = P(l(\boldsymbol{X})|\boldsymbol{X},\boldsymbol{W},\boldsymbol{\beta})P(\boldsymbol{W}|\boldsymbol{\tau},\boldsymbol{\upsilon})P(\boldsymbol{\beta}|\mu,\lambda) \quad (7)$$

where $P(\boldsymbol{W}|\boldsymbol{\tau},\boldsymbol{\upsilon}) = P(\boldsymbol{W}|\boldsymbol{\alpha})P(\boldsymbol{\alpha}|\boldsymbol{\tau},\boldsymbol{\upsilon})$ and $P(\boldsymbol{\beta}|\mu,\lambda) = P(\boldsymbol{\beta}|\boldsymbol{\rho})P(\boldsymbol{\rho}|\mu,\lambda)$. Substituting (3)–(6) to (7), one can iteratively update Ξ by following the gradient descent $\partial P(l(\boldsymbol{X})|\boldsymbol{X},\Xi)/\partial\Xi$. For an updated Ξ, one can update the optimal parameters $\boldsymbol{W}^* = \text{argmax}P(\boldsymbol{W}|\boldsymbol{\tau},\boldsymbol{\upsilon})$ and $\beta^* = \text{argmax}P(\boldsymbol{\beta}|\mu,\lambda)$. The process runs iteratively until reaching some convergence conditions.

Finally, for a new sample $\boldsymbol{x}$, its label can be predicted by (2) with the confidence

$$P(l(\boldsymbol{x}) = c|\boldsymbol{X},\boldsymbol{W}^*,\boldsymbol{\beta}^*) = E_\epsilon\left[\prod_{i\neq c}\Phi(\epsilon + (\boldsymbol{w}_c^* - \boldsymbol{w}_i^*))^{\mathrm{T}}\boldsymbol{K}(x)\boldsymbol{\beta}^*\right],$$

where E_ϵ is the expectation along the variable ϵ.

Compared with SVMs, RVMs offer a number of merits, as an RVMs can be interpreted as an extension of an SVM based on the Bayesian optimal principle [31].

- First, an RVM can classify multiple-label data simultaneously, while an SVM is an intrinsic binary classifier. Hence, in order to recognize multiple classes, an SVM needs to be implemented in a pairwise manner, i.e one-versus-all or one-versus-one. Hence, the number of classifiers increases at least proportionally to the number of classes.
- Second, the output of an RVM is the probability of a sample belonging to a class, thus providing a prediction confidence. An SVM in contrast can only tell if a point is on the left or right of the decision boundary, i.e. true-or-false prediction. Hence, when an SVM is used for multiple classes, there could be a situation that an SVM cannot make a decision. For example, three pairwise SVMs predict three different labels for a sample that can belong to one of three classes. Furthermore, voting mechanism does not yield authentic prediction probability.
- Third, when using an SVM, one must confront the over-fitting problem, i.e. the decision boundary fits perfectly to the training data but yields poor results on the testing. Therefore, the regularization parameter for an SVM must be tuned empirically, usually through a number of cross validations comparing all parameters. This tuning is burdensome especially in a multiple classes case. No optimal parameter is guaranteed though. The principle of an RVM, in contrast, is established based on an assumption that only a small subset of the training dataset is important to construct the boundary while the rest is safe to discard. The weights of these vectors, i.e. the relevance vectors, are optimized automatically based on the Bayesian principle. Thus an RVM avoids the over-fitting problem in an elegant way without any tuning requirements.
- Fourth, the sparsity of an RVM is obtained through the Bayesian principle, while sparsity of an SVM is obtained through a penalization term, i.e. typically by constraining the L1 or L2 norm of the parameters. Hence, the number of relevance vectors in an RVM is significantly smaller than that of an SVM. Thus, the prediction process of RVM is more efficient and faster than an SVM whilst maintaining comparable generalization error.
- Lastly, the kernel in an RVM does need not to be positive definite, while the kernel in SVM must satisfy the Mercer's condition [31]. In this work though, since we used the log-Euclidean with Gaussian kernel, the Merce condition is satisfied and this is not a clear advantage. However, an RVM can avoid this constraint if one wants to use other Riemannian distance metrics directly with the Gaussian kernel.

The principle disadvantage of an RVM is that it is computationally expensive for training, since the optimization function is non-convex. The training time of an RVM is often longer than an SVM for a binary class case. However, for multiple classes, this difference is not noticeable, considering that a heavy cross-validation process of an SVM is needed in order to obtain a good performance. For testing, however, an RVM is significantly faster than an SVM, which we are more concerned about.

3.2. Proposed approach

3.2.1. Extracting low level features. In order to construct the covariance matrix, we first need to extract certain low level features. In this work, we investigate two ways to extract low level features:

- Using the preprocessed signal from selected channels directly, i.e. $X \in \mathbb{R}^{m\times n}$, where m is the number of channels, and n is the number of samples in one epoch. This low level feature has been used successfully in motor imagery, such as in [32].
- In order to exploit information from the frequency domain, we also extract wavelet coefficients using the Morlet wavelet for each channel. The features vector then is a concatenation of the channel index i_{Chn}, the raw data, and the wavelet coefficients of each channel, i.e. $X \in \mathbb{R}^{d\times L}$, where $d = n_{wc} + 2$, n_{wc} is the number of wavelet coefficients at one instant time sample, and $L = nm$. Notice that although the variance of the channel index (diagonal element) is equal for all trials, its correlations to other features (off diagonal elements) vary among trials. Doing so allows us incorporating both spatial and spectral correlation while keeping the size of covariance matrix small, which also ensures that the covariance matrix is well defined, i.e. positive definite.

3.2.2. Main procedure. The main procedure of the proposed method is described in figure 1, which includes the following steps.

- In the preprocessing step, we apply frequency and spatial filters to select the most corresponding frequency band and channels.
- Low level features are then extracted as described in section 3.2.1, and covariance matrices are computed. The mean of the covariance matrices is obtained from the training set as described in algorithm 1.

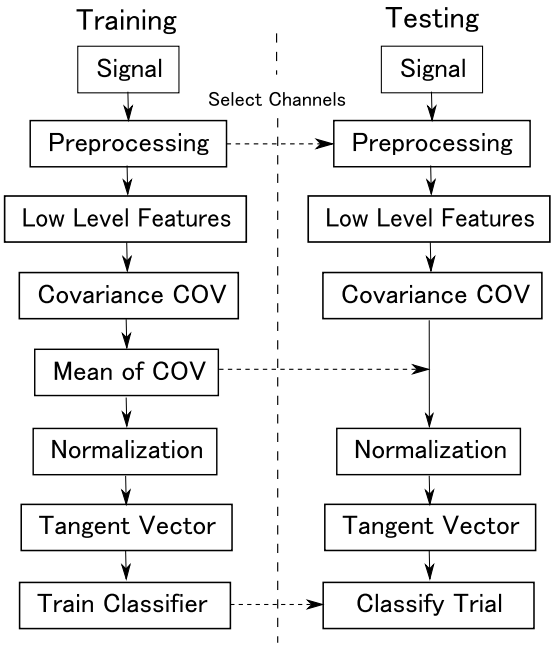

Figure 1. Main procedure of the proposed method.

- This geometric mean matrix obtained from the training set is then used to normalize the whole dataset and compute the tangent vectors according to (1). An RVM is used to train and classify the tangent vectors.

4. Experiments procedure

4.1. Main procedure.

In this work, 15 healthy subjects (S1–15, 11 males and 4 females, ages 22–32) performed three different types of imagined speech, namely imagined speech of *short words*, *long words* and *vowels*. All subjects were right-handed except subject S13. The aim was to investigate the mechanisms of imagined speech and evaluate the suitability of each group for BCI applications. The group of short words included the words 'in', 'out' and 'up', while the group of long words consisted of 'cooperate' and 'independent'. These words were chosen in order to evaluate the effect of the meaning and the complexity of the words. In order to evaluate the effect of the sound, three phonemes were used, namely /a/, /i/ and /u/. Finally, in order to further analyze the effect of the complexity, i.e. the length and the different sounds, we performed an additional experiment where the subjects had to imagine either one of the short words ('in') or one of the long words ('cooperate').

During the experiments, the subjects were instructed to pronounce these words internally in their minds and avoid any overt vocalization or muscle movements. The subjects were receiving instructions about the desired word/phoneme based on visual cues from a computer monitor. The experimental setup is shown in figure 2, while figure 3 shows the experimental procedure which is described in more detail below. The experimental protocol was approved by the ASU IRB (Protocols: 1309009601, STUDY00001345) and each participant signed an informed consent form before the experiment.

Figure 2. Experimental setup. The subject is wearing the cap with EEG electrodes and looks at a monitor a few inches away. The monitor shows the task that the subject must execute. In this illustrative figure, the subject was imagining pronouncing the word 'in'.

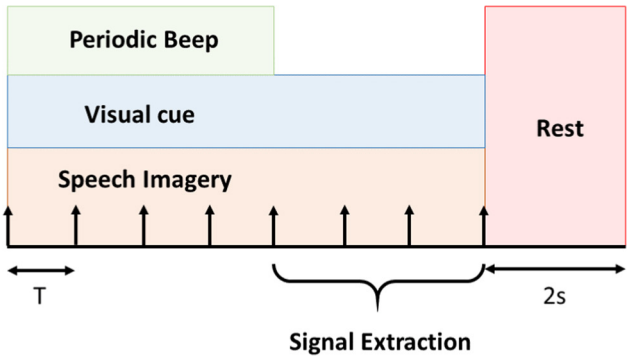

Figure 3. Experimental procedure. The vertical arrows represent the time instants where the subject was expected to perform speech imagery, and T denotes the rhythm period.

Each subject performed one to three sessions of imagined speech. Each session corresponded to one of the mentioned groups of speech imagery, e.g. three short words, and was conducted approximately in 1 h. A single experimental session was comprised of 100 trials per word or sound, which were shown randomly. During each trial, the subject would hear a beep sound that was repeated at period T. This helped create the rhythm that subjects should imagine pronouncing the words or phonemes. In more detail, the beep sound appeared firstly when the trial started and was repeated four more times. At the beginning of the trial, the subject was also prompted with a visual cue indicating the desired word to be imagined. The cue lasted for $7 \times T$ s. The subject was instructed to perform speech imagery at each beep sound and continue at the same rhythm until the visual cue disappeared. This resulted in the subject performing speech imagery for an additional three periods after the last beep sound. Finally, the trial ended with a rest period of approximately 2 s where no cue and no sounds were present. For short words and vowels, the period was $T = 1$ s, while for long words the period was $T = 1.4$ s. In the case of comparing between a short and a long word, $T = 1.4$ s was also chosen to render the comparison more accurate. The values for T were chosen empirically based on how long it would take the subjects to pronounce the words overtly.

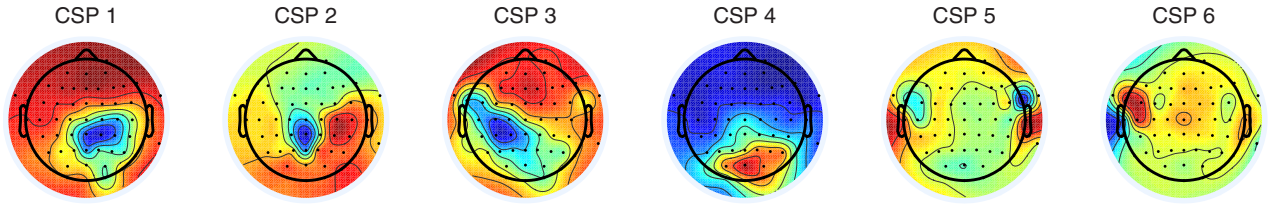

Figure 4. The first 6 CSP patterns for subject S3 during speech imagery of short words. These patterns correspond to the first CSP method which aims to distinguish between the mental task of speech imagery in general, i.e. irrespectively of the imagined word, and the resting condition. Here, only those patterns that increase the variance of speech imagery and reduce the one of resting are presented.

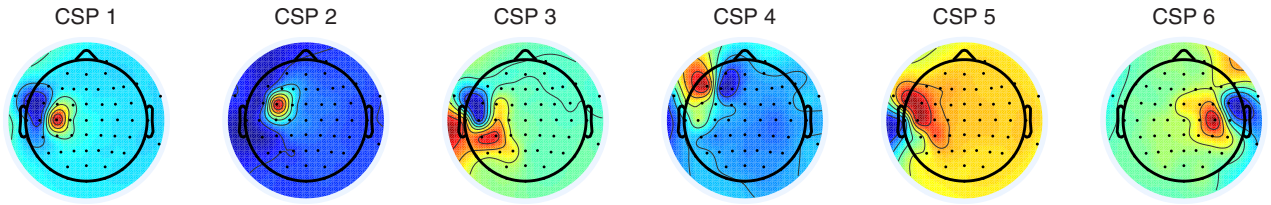

Figure 5. The first 6 CSP patterns for subject S3 during speech imagery of short words. These patterns correspond to the second CSP method which aims to differentiate between each of the individual classes/words. The method ranks the CSP patterns based on their mutual information. Here, only the first components are presented because they contain the most information about the mental task of speech imagery.

This experiment procedure is similar to the ones presented in D'Zmura [11] and Brigham [13]. It is important to notice that in our work, as in those experiments, the analyzed signals correspond to a segment of the experiment where only a visual cue is present and any auditory cues have been stopped. The aim is to avoid any possible evoked brain activity related to the sound that may lead to misinterpretation of the data. Furthermore, we are interested in classifying different words or sounds imagined in the same period, but not between different rhythms as investigated by Deng *et al* [16] and their related works.

4.2. Data acquisition and conditioning

The EEG signals were acquired using a BrainProducts ActiCHamp amplifier system from 64 electrodes placed according to the 10/20 international system [33]. The data were recorded at 1000 Hz and later they were downsampled at 256 Hz. During preprocessing, a 5th order Butterworth bandpass filter between 8–70 Hz was applied to remove any low-frequency trends in the data as well as possible artifacts related to EMG activity. A notch filter at 60 Hz was also applied in order to remove line noise. Finally, an electro-oculogram artifact removal algorithm [34] was applied on the data to eliminate any eye blinking or eye movement artifacts.

5. Data analysis and results

5.1. Data analysis

We first analyzed the data to show that the collected signals were corresponding to speech imagery, and investigated several techniques to select the corresponding channels. The first method was based on CSPs [35–37], as has been done in many related works. The CSP method was applied in two different ways. In the first case, we simply performed a binary class CSP analysis, where the first class contained signals during mental imagery and the second one contained signals during resting in the trial. The regularized CSP toolbox provided by Lotte *et al* [38] is used to perform CSP in this work. In the second method, the multi class CSP [39] was applied. This technique first performs independent component analysis (ICA) to obtain orthogonal channels, and selects the channels with the highest mutual information with the corresponding labels.

Figure 4 shows the first six CSP patterns for subject S3 during imagination of short words using the first CSP method. The corresponding CSP analysis aims to distinguish between the mental task of speech imagery in general, i.e. irrespectively of the imagined word, and the resting condition. In this binary analysis, the first and the last CSP patterns correspond one-by-one to the CSP filters that increase the variance of the speech imagery task and the resting condition, respectively. Since the aim is to reduce the number of channels for the feature extraction process, only those patterns that are related with the speech imagery task and not with the resting phase are considered and shown in figure 4.

On the other hand, figure 5 shows the CSP patterns for the same subject during short word imagination but following the second CSP method. As explained, the corresponding multi-class CSP patterns produced by this method are ranked based on their mutual information scores with each mental task from highest to lowest. The first six of those hence contain the most information about the speech imagery and are selected to show in figure 5.

By examining the results of figure 4 and figure 5, we conclude that the brain activity during the experiment was indeed concentrated almost exclusively on the left frontal, middle and parietal sides of the brain which lie over Broca's area, the motor cortex and Wernicke's area. These areas are involved in speech production and recognition, as also acknowledged

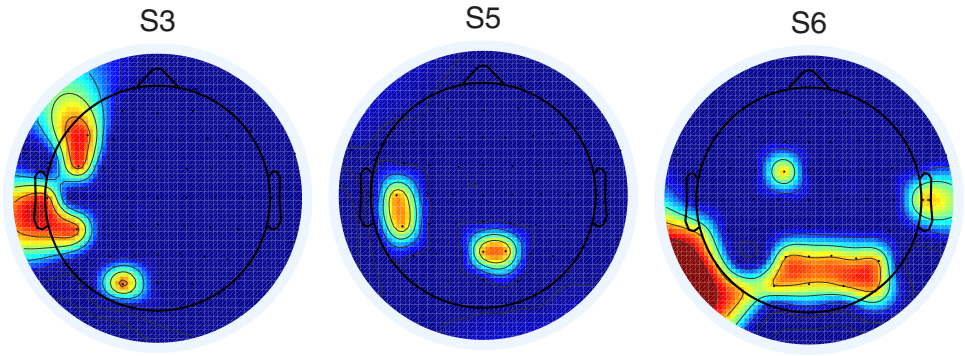

Figure 6. Scalp map of the thresholded autocorrelation score of the channels for subjects S3, S5, S6 performing short words imagery.

in the related literature [12, 40, 41]. This is more evident in figure 5 (CSP1–CSP5) compared to figure 4 (CSP3 and CSP6). More specifically, in figure 4, we notice that some components (CSP2, CSP4) relate more to areas in the visual cortex. This is to be expected since during the main trial a visual cue was presented to the users while in the rest period the cue was absent. Thus, the CSP analysis just detects and shows that difference. However, this visual stimulus is present in all trials across all classes, therefore, its effect on the actual classification procedure is negligible.

To further verify the above conclusions, we performed an additional analysis that was based on the autocorrelation of the signals on each channel. In the experiments the subjects were instructed to perform speech imagery at specific time intervals defined by auditory cues at a certain rhythm. The basic premise is that this rhythm should also be visible in the autocorrelation function of the signals. We detected which channels exhibited such periodicity repeatedly across trials by analyzing the fast Fourier transform (FFT) of the autocorrelation function of each channel and scoring those channels whose highest peak in the FFT was close to the frequency of the auditory cue. We picked the ones that were exhibiting such periodicity in 70–80% or more of the trials (depending on the subject). The chosen channels for subjects S3, S5 and S6 are shown in figure 6. As it can be seen, this analysis also points more towards the same areas of the brain as in the CSP methods discussed previously which solidifies the validity of the data.

Among the three methods, the binary CSP provides the highest classification accuracy and is quite consistent in most of the cases. It is also the simplest method as it is unsupervised in our implementation, i.e. all mental tasks are considered as one class, e.g. involving speech imagery, instead of separated classes. The multi-class CSP occasionally yields better results than the first method, depending on the subjects and conditions. Finally, selecting channels based on the autocorrelation method yields the least classification accuracy in our experiment. Hence, in the later discussion, we only report the results using the CSP algorithms.

Furthermore, we also performed a time-frequency analysis on the data. Among the mentioned types of speech imagery, the frequency domain characteristics between a short word and a long word show some noticeable discrepancies. Specifically, figure 7 shows the time–frequency response of channels FC5, FC3, F5, FT7 (Broca's area) and CP5, TP7, CP3 and P5 (Wernicke's area) for subject S14. Concretely, we first obtained the Morlet wavelet transform using a setting of eight octaves with 20 voices during a period of 8 s in each trial, which consists of 2 s before the last beep, 4.2 s of the main trial, and 1.8 s of the pause period, i.e. resting. Then, we took the average of these wavelet magnitudes across all trials for each class.

In figure 7, we observe activity in the frequency range of 60–70 Hz, i.e. close to the line noise frequency of 60 Hz. However, in our signal preprocessing we applied a notch filter with a bandwidth of 3 Hz centered at 60 Hz , i.e. −3dB at 58.5 Hz and at 61.5 Hz. Furthermore, the effect is not noticeable in channels TP7, CP5, CP3 and P5. Thus, it is safe to reject line noise as the source. Similarly, there is also activity in the frequency band of 20–31 Hz at the channels of Broca's area, which can be best observed in channels F5 and FC3. However, in both frequency ranges, i.e. 60–70 Hz and 20–31 Hz, the activity changes overtime and still persists during the resting state. This may indicate idling of the brain at these frequency ranges for the specific subject and channels of interest.

Furtheremore, the activity in Broca's area mainly appears above 20 Hz, whereas that in Wernicke's area appears below 15 Hz. Figure 7 also points out that the brain activity remains high during resting, i.e. last 1.8 s, and decreases during mental imagination. In addition, the activity seems to be more suppressed in the short word case than the long word, especially in the high frequency band (31–70 Hz) and the channels of the Broca's area.

5.2. Classification results

To evaluate the proposed method, we performed a 10-fold cross-validation procedure. The training and testing sets were partitioned randomly. For short words and vowels, we classified the data across three classes with 90% (270 trials) of the datasets used for training and 10% (30 trials) for testing. For long words and the comparison between a short and a long word, 80% (160 trials) of data are used for training and 20% (40 trials) for testing.

For short words and vowels, we extracted 4 s for each trial. The first 3 s correspond to the expected speech imagery. A period of 1 s after that was further added in order to capture mental activity that the subject might still perform after the visual cue disappeared. The total 4 s time period is further divided into three epochs of 2 s with a 1 s overlap. For long words and the short-versus-long word comparison, we extracted 4.5 s per trial. This interval was also divided into three epochs of 2 s with a 1.25 s overlap.

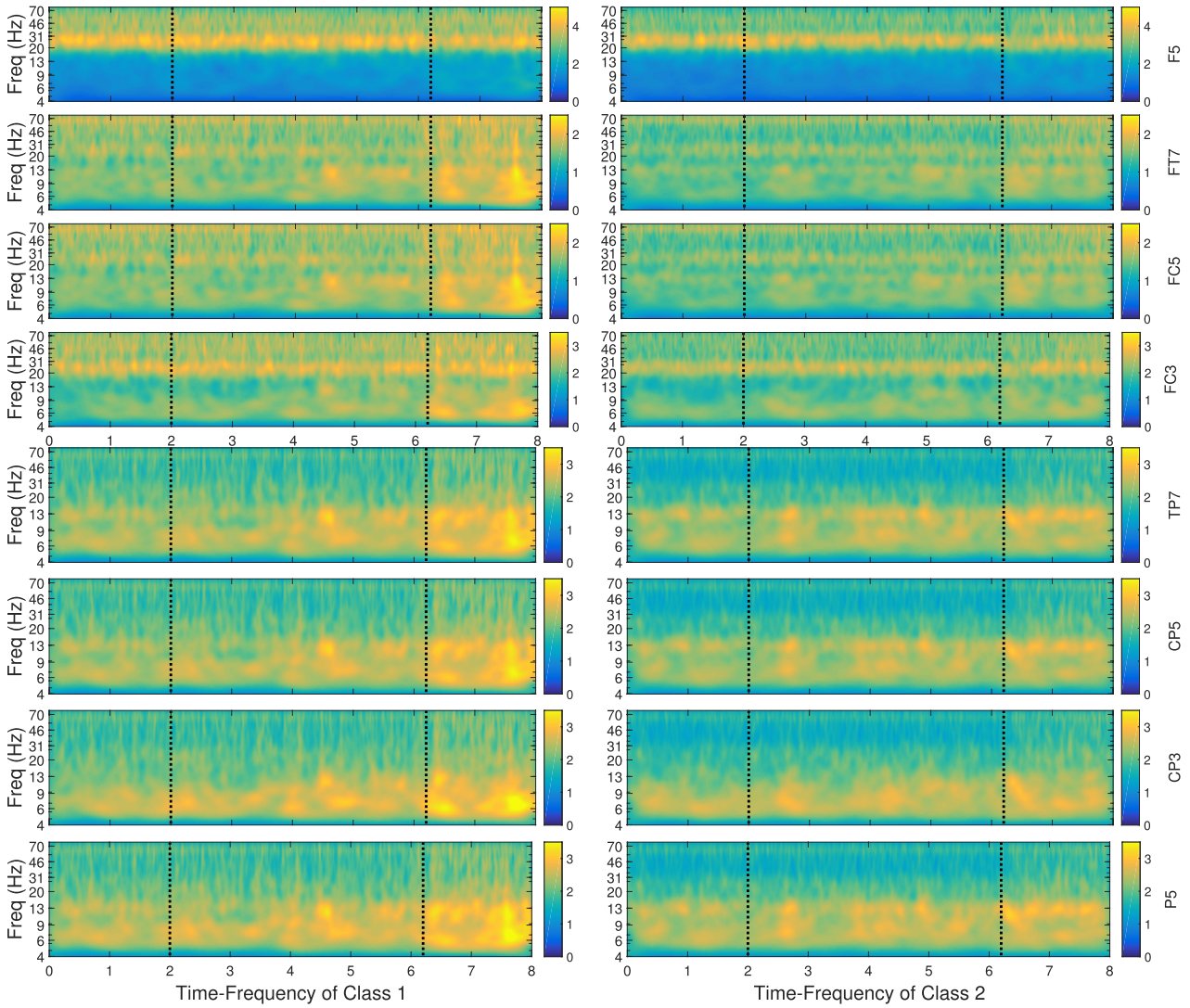

Figure 7. Time-frequency response of channels F5, FT7, FC5, FC3 (Broca's area) and TP7, CP5, CP3, P5 (Wernicke's area) corresponding to speech imagery of a long word (class 1) and a short word (class 2) for subject S14. The left dotted line corresponds to the last beep, and the right denotes the end of the main trial in accordance to our experimental protocol.

The number of CSP components was chosen empirically and it varied between 6 and 14 depending on the subjects and testing conditions. In most of the cases, the binary CSP method is used to select the channels. In the last test described later, we occasionally found that the multi-class CSP would yield better results. In all cases, multiple Gaussian kernels are used for the mRVM. The multi kernel parameters were typically set to $[20, 10, 1, 0.1, 0.05, 0.025, 0.01]$ with small adjustments for each subject and each case.

When classifying short words, vowels, and long words, we found that incorporating frequency information did not provide any improvement, so that only the signals spatially filtered by CSP were used to compute the covariance matrix (method 1). In contrast, when classifying short versus long words, combination of the two features by simply concatenating the tangent vectors from two covariance matrices (method 2) can improve the results in most subjects. This suggests that the difference in the complexity of the words could create discriminative features across frequency bands. To extract the second type of low level features, Morlet wavelet transform using the function *cwt* provided by Matlab was used. The number of octaves and number of voices for Morlet wavelet transform are set to 8 and 10 respectively, which yields totally 80 $(= 8 \times 10)$ scales. Only the scales in the range $[8, 40]$ are used to construct the low level feature vector, which in turn yields the low level feature vector in the dimension of 35 $(=$ 33 subands $+1$ raw signal $+1$ channel index). Since the dimension of the tangent vector extracted from this covariance matrix is high, we apply PCA to reduce its dimension into 15–30 components. The highest accuracy in this case is obtained by using the multi-class CSP algorithm with 12–14 CSP components.

The classification results of all cases are presented in figure 8 and table 1, which show the mean, minimum and maximum values for each subject participating in each group. The two last groups in figure 8 represent the results of classifying a short versus a long word using the first set of low level features (method 1), and the combination between the two kinds of features mentioned previously (method 2). To compare the classification accuracy between groups of different

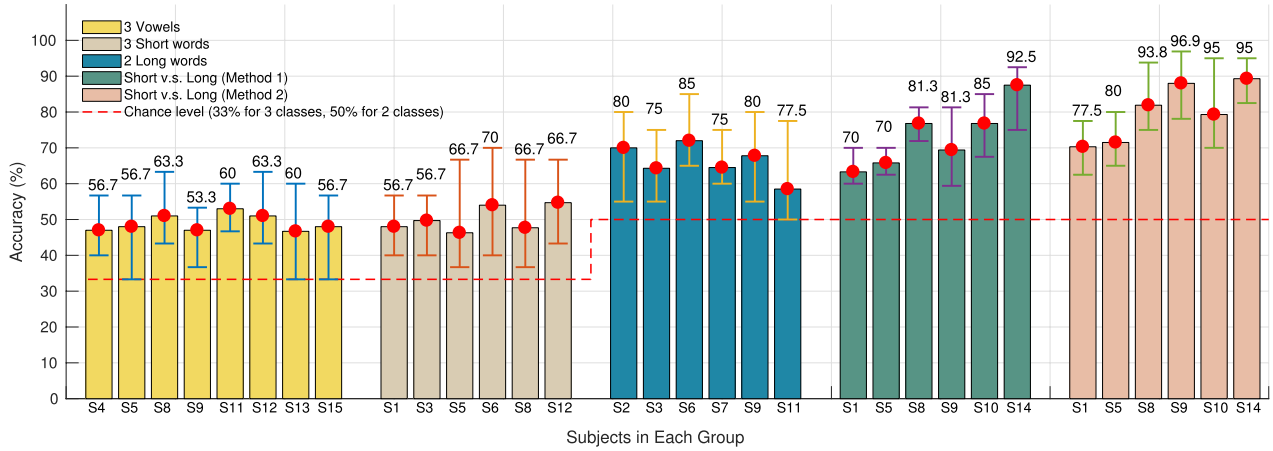

Figure 8. Mean, minimum and maximum classification accuracy (%) for different subjects in each mental task.

Table 1. Mean ± Std of the accuracy (%) and kappa values for all subjects and speech imagery types.

(a) Vowels

Subject	S4	S5	S8	S9	S11	S12	S13	S15
Accuracy	**47.0** ± 4.6	**48.0** ± 7.2	**51.0** ± 6.7	**47.0** ± 5.5	**53.0** ± 4.0	**51.0** ± 6.3	**46.7** ± 8.2	**48.0** ± 7.2
κ value	**0.21** ± 0.07	**0.22** ± 0.11	**0.27** ± 0.10	**0.21** ± 0.08	**0.30** ± 0.06	**0.27** ± 0.09	**0.20** ± 0.12	**0.22** ± 0.11

(b) Short words

	S1	S3	S5	S6	S8	S12
Accuracy	**48.0** ± 6.1	**49.7** ± 5.5	**46.3** ± 8.2	**54.0** ± 9.1	**47.7** ± 9.8	**54.7** ± 6.9
κ value	**0.22** ± 0.09	**0.25** ± 0.08	**0.20** ± 0.12	**0.31** ± 0.14	**0.22** ± 0.15	**0.32** ± 0.10

(c) Long words

	S2	S3	S6	S7	S9	S11
Accuracy	**70.0** ± 7.8	**64.3** ± 6.6	**72.0** ± 0.6	**64.5** ± 5.5	**67.8** ± 6.8	**58.5** ± 7.4
κ value	**0.40** ± 0.16	**0.29** ± 0.13	**0.44** ± 0.12	**0.29** ± 0.11	**0.36** ± 0.14	**0.17** ± 0.15

(d) Short versus long words (method 1)

	S1	S5	S8	S9	S10	S14
Accuracy	**63.3** ± 2.9	**65.8** ± 3.1	**76.8** ± 3.0	**69.4** ± 7.5	**76.8** ± 6.2	**87.5** ± 5.5
κ value	**0.27** ± 0.06	**0.32** ± 0.06	**0.54** ± 0.06	**0.39** ± 0.15	**0.54** ± 0.12	**0.75** ± 0.11

(e) Short versus long words (method 2)

	S1	S5	S8	S9	S10	S14
Accuracy	**70.3** ± 5.5	**71.5** ± 5.0	**81.9** ± 6.5	**88.0** ± 6.4	**79.3** ± 7.9	**89.3** ± 3.5
κ value	**0.41** ± 0.11	**0.43** ± 0.10	**0.64** ± 0.13	**0.76** ± 0.13	**0.59** ± 0.15	**0.79** ± 0.07

number of classes, we further compute the κ value, which is defined as

$$\kappa = 1 - \frac{1 - P\%}{1 - C\%},$$

where $P\%$ is the prediction accuracy, and $C\%$ is the chance level, i.e. $C\% = 50\%$ for two classes. The mean, minimum and maximum κ values for each subject participating in each group are presented in figure 9 and table 1. The averaged classification accuracies for all subjects in each group are reported in table 2, which is well above chance level.

6. Discussion

6.1. Comparing with previous approaches in the literature

In order to evaluate the performance of the proposed approach, we further compare the results with several works in the field using the datasets of vowels, short words and shot-long words.

We first compare with the standard baseline approach used in BCI applications, which commonly includes a CSP filter, followed by extracting log-variance as the feature and an LDA classifier [42]. To classify three different vowels or short words, in the training step, for each pairwise class, such as

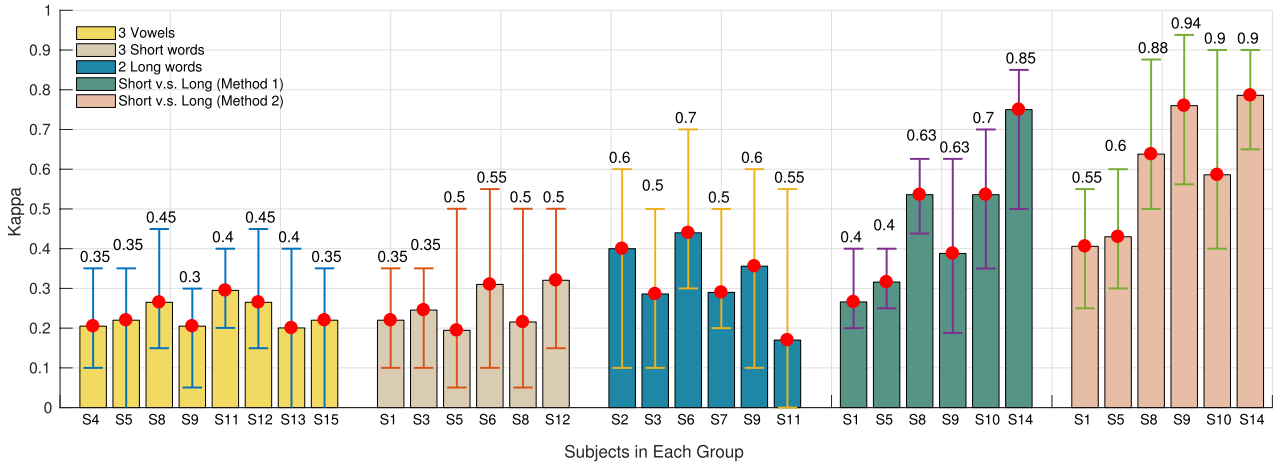

Figure 9. Mean, minimum and maximum of kappa (κ) value for different subjects in each mental task

Table 2. Average of mean $\pm$ Std of the accuracy (Acc) and kappa (κ) values.

Group	Participants	Acc (%)	κ
Vowels	S4,S5,S8,S9, S11,S12,S13,S15	49.0 ± 2.4	0.23 ± 0.04
Short (S) Words	S1,S3,S5,S6,S8,S12	50.1 ± 3.5	0.25 ± 0.05
Long (L) Words	S2,S3,S6,S7,S9,S11	66.2 ± 4.8	0.32 ± 0.10
S & L (Method 1)	S1,S5,S8,S9,S10,S14	73.3 ± 8.9	0.47 ± 0.18
S & L (Method 2)	S1,S5,S8,S9,S10,S14	80.1 ± 8.0	0.60 ± 0.16

(1–2), (2–3), (3–1), we constructed a CSP filter. The log variance of the CSP filtered channels is then used as the feature. We tested with two to six CSP components, which yield the feature vector in dimension of 4 to 12 respectively. Next, the mean of the feature vectors in the training set is used to whiten the dataset, i.e. subtracting the data set by the mean, before calibrating the LDA classifier. The LDA's regularization parameter is tested from 0.05 to 0.95 with 0.05 step. Hence, this training process yields totally three classifiers of a triplet (CSP matrix, mean vector, LDA). In the testing process, we applied three classifiers to each testing data point, i.e. applying CSP, computing log-variance, subtracted by the feature vector mean generated from the training process, and applying LDA classifier. In our tests, a number of CSP components between six to eight and a regularization parameter of 0.1 yield best results in most subjects. The same procedure to extract the time epoch from the recorded EEG as in our proposed approach is used. That is, a trial with 4 s is divided into three epochs, each has 2 s with 1 s overlap. The label of the tested trial is assigned by voting mechanism. Since each trial has three epochs and each epoch is tested against three classifiers, there is total nine predicted labels for each trial. The label that appears most frequently and at least four times is assigned to that trial. The results are reported in table 3 by taking the best mean values among the mentioned parameter set. It can be seen that the accuracy is even below the chance level.

In [13], the authors study the dataset provided by [11] in order to classify two syllables, namely /ba/ and /ku/, whereas the experimental procedure in [11] is very similar to ours. According to their report [13], classification using the full dataset did not provide accuracy greater than chance. Thus, the Hurst exponent was used to select good trials. In our attempt to implement this method, the ICALAB toolbox [43] is used to perform ERICA algorithm, and the threshold (0.70–0.76) on Hurst exponent is set to select the independent components as in their report. In our case, there were many trials that were rejected from the dataset due to all independent components being rejected. For example, for subject S13, only 33, 23 and 28 trials were deemed useful for each vowel, respectively. In our opinion, although the use of the Hurst exponent provides an automated method for rejecting bad channels/ICs which is critical in the BCI field, it might also be too restrictive for a BCI application for two reasons. First, especially compared to the number of 100 trials per class, the number of valid trials is too small for creating a training and a testing dataset that will result in a robust classifier and an accurate validation procedure. Second, this approach could be problematic for a user in a real-time scenario as most of the time the user would not have control of the system. As feedback is deemed very important during learning of a BCI system [44], this strict rejection would create significant issues during operation because the system would not respond. Our method on the other hand, does not involve any rejection but rather EEG artifact correction which results in classification of EEG signals without interruptions.

We also compare our results with the method proposed by DaSalla *et al* [14]. In their approach, the first 500 ms of each trial are used to extract the features. Specifically, for each pairwise classification, four CSP channels are computed from the training sets and sequentially used to decompose the training and testing time series. The final feature vectors are obtained by concatenating the time series of the four channels without

Table 3. Comparing mean ± std, min ÷ max accuracy between different methods for speech imagery.

(a) Comparing methods for Vowels data set

Subject	S4	S5	S8	S9	S11	S12	S13	S15
Log + LDA	29.3 ± 9.9 20.0 ÷ 50	30.6 ± 7.9 20.0 ÷ 43.3	34.6 ± 11.8 16.6 ÷ 60	40.3 ± 10.4 13.3 ÷ 46.6	31.0 ± 8.5 20.0 ÷ 46.6	37.3 ± 7.1 30.0 ÷ 53.3	37.5 ± 8.1 26.6 ÷ 53.3	31.0 ± 12.7 10.0 ÷ 50.0
Dasalla *et al* [14]	32.3 ± 7.4 26.7 ÷ 50	36.3 ± 2.9 33.3 ÷ 43.3	36.7 ± 9.2 30.0 ÷ 60	34.7 ± 7.7 30.3 ÷ 53.3	33.7 ± 8.7 23.3 ÷ 53.3	41.7 ± 5.7 36.7 ÷ 56.7	38.7 ± 7.6 30.3 ÷ 56.7	31.0 ± 7.4 23.3 ÷ 46.7
Min *et al* [45]	35.8 ± 4.1 30.0 ÷ 43.3	39.4 ± 4.6 30.0 ÷ 46.7	46.5 ± 5.6 36.7 ± 56.7	36.1 ± 4.4 26.6 ÷ 43.3	36.0 ± 7.1 30.0 ÷ 50.0	39.3 ± 9.4 30.0 ÷ 60.0	34.3 ± 6.5 23.3 ÷ 43.3	34.0 ± 8.3 23.3 ÷ 46.7
Tangent + ELM	41.0 ± 13.3 23.0 ÷ 66.7	44.6 ± 11.2 30.3 ÷ 66.7	45.3 ± 8.9 30.0 ÷ 56.7	46.0 ± 5.1 36.7 ÷ 53.3	43.3 ± 7.9 33.3 ÷ 53.3	48.6 ± 8.9 36.7 ÷ 60.0	45.7 ± 7.2 36.7 ÷ 63.3	46.7 ± 7.5 36.7 ÷ 60.0
Tangent + RVM	**47.0** ± 4.6 40.0 ÷ 56.7	**48.0** ± 7.2 33.3 ÷ 56.7	**51.0** ± 6.7 43.3 ÷ 63.3	**47.0** ± 5.5 36.7 ÷ 53.3	**53.0** ± 4.0 46.7 ÷ 60	**51.0** ± 6.3 43.3 ± 63.3	**46.7** ± 8.2 33.3 ÷ 60.0	**48.0** ± 7.2 33.3 ÷ 56.7

(b) Comparing methods for short words data set

Subject	S1	S3	S5	S6	S8	S12
Log + LDA	39.6 ± 7.6 26.6 ÷ 53.3	32.6 ± 4.9 26.6 ÷ 43.3	27.7 ± 9.8 20.0 ÷ 50	33.7 ± 6.9 23.3 ÷ 43.3	43.3 ± 7.0 36.6 ± 53.3	27 ± 10.8 13.3 ÷ 43.3
Dasalla *et al* [14]	42.3 ± 8.2 33.3 ÷ 56.7	38.3 ± 5.3 33.3 ÷ 50.0	35.3 ± 8.3 30.0 ÷ 56.7	36.0 ± 5.2 33.3 ÷ 50.0	38.3 ± 6.1 33.3 ÷ 53.3	41.33 ± 6.7 33.3 ÷ 53.3
Min *et al* [45]	41.0 ± 5.5 46.7 ÷ 56.7	42.3 ± 8.0 26.7 ÷ 56.7	**48.3** ± 7.2 36.7 ÷ 60.0	32.3 ± 8.0 23.3 ÷ 43.3	34.7 ± 5.9 26.7 ÷ 46.7	49.0 ± 6.7 36.7 ÷ 56.7
Tangent + ELM	44.6 ± 10.3 33.3 ÷ 60.0	45.3 ± 7.4 33.3 ÷ 56.7	43.4 ± 7.7 30.0 ÷ 56.7	46.3 ± 8.1 36.7 ÷ 56.7	45.0 ± 8.5 30.0 ÷ 56.7	**55.0** ± 9.8 40.0 ± 70.0
Tangent + RVM	**48.0** ± 6.1 40.0 ÷ 56.7	**49.7** ± 5.5 40.3 ÷ 56.7	46.3 ± 8.2 36.7 ÷ 66.7	**54.0** ± 9.1 40 ÷ 70	**47.7** ± 9.8 36.7 ÷ 66.7	54.7 ± 6.9 43.3 ÷ 66.7

(c) Comparing methods for short versus long words data set

Subject	S1	S5	S8	S9	S10	S14
Log + LDA	50.5 ± 14.8 30.0 ÷ 72.5	59.5 ± 5.7 52.5 ÷ 70.0	36.9 ± 15.9 21.9 ÷ 71.9	74.1 ± 16.6 31.3 ÷ 87.5	64.3 ± 23.0 20.0 ÷ 80.0	78.5 ± 6.3 70.0 ÷ 90.0
Dasalla *et al* [14]	61.5 ± 12.0 50.0 ÷ 85.0	61.5 ± 8.8 50.0 ÷ 80.0	62.5 ± 8.3 50.0 ÷ 81.3	58.1 ± 7.2 50.0 ÷ 75.0	66.0 ± 11.5 50.0 ÷ 85.0	54.5 ± 13.2 45.0 ÷ 90.0
Min *et al* [45]	51.0 ± 8.4 40.0 ÷ 65.0	59.5 ± 6.4 50.0 ÷ 70.0	59.4 ± 11.5 43.8 ÷ 81.3	51.9 ± 6.6 43.8 ÷ 68.8	61.0 ± 9.7 45.0 ÷ 75.0	54.0 ± 6.1 50.0 ÷ 70.0
Tangent + ELM	**73.5** ± 8.2 60.0 ÷ 85.0	**70.0** ± 6.2 60.0 ÷ 80.0	80.6 ± 13.2 62.5 ÷ 93.8	72.5 ± 12.2 43.7 ÷ 87.5	75.5 ± 6.8 65.0 ÷ 85.0	85.5 ± 6.8 75.0 ÷ 95.0
Tangent + RVM (Method 1)	63.3 ± 2.9 60.0 ÷ 70.0	65.8 ± 3.1 62.5 ÷ 70.0	76.9 ± 3.0 71.8 ÷ 81.3	69.4 ± 7.5 59.4 ÷ 81.3	76.8 ± 6.2 67.5 ÷ 85.0	87.5 ± 5.5 75.0 ± 92.5
Tangent + RVM (Method 2)	70.3 ± 5.5 62.5 ÷ 77.5	**71.5** ± 5.0 60.0 ÷ 80.0	**81.9** ± 6.5 75.0 ÷ 93.8	**88.0** ± 6.4 78.1 ÷ 96.9	**79.3** ± 7.7 70.0 ÷ 95.0	**89.3** ± 3.5 82.5 ± 95.0

further processing. The SVM with a radial basis functions kernel is used to classify the data. In our implementation, the sampling frequency is 256 Hz, which yields the feature vector of dimension $512 (= 4 \times 128)$. We use 90% of the dataset for training and 10% for testing, and perform 20 cross validations. The functions *svmTrain* and *svmclassify* provided by Matlab are used, where the penalizing parameter takes the values in the set $\{0.01, 0.1, 1, 10, 100\}$ and the Gaussian kernel parameter takes the values in the range from 0.1 to 40 in increments of 0.2. The label of each trial is decided based on voting from each pairwise trained SVM classifiers. The results of applying their method on our data is reported in table 3 by taking the best mean values among the mentioned parameter set. However, the accuracy is low, in certain cases being even below chance level. This may be due to the difference between the two experimental procedures. In their experiments, subjects were instructed to imagine vocalizing and mouthing the vowels. As mentioned by Brigham *et al* [13], the set of instructions proposed by Dasalla *et al* may be more related to motor imagery, where the CSP yields best results. In contrast, our experiment emphasizes on imagery of the sound, which is more related to speech imagery.

We also implemented the method proposed by Min *et al* [45], which yields excellent results on their dataset. In their approach, each 3 s imagination trial is bandpass filtered in $[1, 70]$ Hz and partitioned into 30 segments of 0.2 s with 0.1 s overlap. Four features, namely mean value, variance, standard deviation, and skewness, are extracted from each segment. Concatenating of these features from a total of 60 channels yields the final feature vector of dimension 240 ($= 4 \times 60$). Each segment is then treated as an independent sub-trial. Sparse regression is then utilized to perform feature selection and reduce the dimension. The extreme learning machine (ELM) classifier is used to learn the boundary decision from the training set and classify the test samples. The final label of the trial is then determined by the labels mostly appearing from all sub-trials. In our implementation, the function *lasso* provided by Matlab is implemented to perform sparse regression, where the regularization parameter λ takes the values from 0 to 0.2 in increments of 0.01. The optimal λ is selected based on the smallest mean square approximation error *MSE* returned, as suggested in the original paper. The multi-class ELM[4] with linear kernel proposed by [46] is used in our implementation, since the linear kernel is the most suitable in term of sensitivity and specificity reported by Min *et al* [45]. The regularization parameter for ELM is set from 0 to 5 by increments of 0.5. The slight difference is that we used the multi-class version of ELM while Min *et al* only considered pairwise classification. The results are reported in table 3 by taking the best mean values among the mentioned parameter set using totally 20 cross validations.

To further verify the performance of Riemannian feature, we also replace the RVM by the ELM algorithm. Concretely, the tangent vectors are extracted using the same procedure described in figure 1, and the ELM with Gaussian kernel proposed by [46] is used to classify the features. The regularization parameter is tested from 0.01 to 5 with step 0.25, and the highest results are reported in table 3. In our test, this method also yields better results than other methods, which proves the effectiveness of the Riemannian feature. The training time of ELM is much smaller than that of the RVM. However, the result comparing with the RVM method is not stable, as sometimes the accuracy is very low, thus yielding a large standard deviation. This instability could be due to the principle that ELM selects the weights of the first layer of the neural network randomly. The result of RVM is more stable, and hence preferred in our approach. We further performed t-tests on the mean accuracies reported in table 3 (a), (b) and (c) at the 5% significance level. For the vowels, the RVM ($\mu = 49.0, \sigma = 2.4$) performed better than the ELM classifier ($\mu = 45.2, \sigma = 2.3$); ($t(7) = 3.48, p = 0.01$). For the short words, the RVM ($\mu = 50.1, \sigma = 3.5$) also outperformed the ELM classifier ($\mu = 46.6, \sigma = 4.2$); ($t(5) = 3.26, p = 0.02$). For the short versus long word case though, the performance of the RVM ($\mu = 73.3, \sigma = 8.9$) and the ELM classifier ($\mu = 76.3, \sigma = 5.8$) was quite equivalent; ($t(5) = 1.39, p = 0.16$).

[4] Available from the author Guang Bin Huang's website: http://www.ntu.edu.sg/home/egbhuang/elm_kernel.html.

Gonzalez-Castaneda *et al* [22] proposed a method to classify imagery speech which is reported with very high accuracy. However, their experiment protocol is problematic. According to their report [22], 'imagined pronunciation of each word was repeated 33 times in succession', and the subject was told which word to imagine before each block. This experimental procedure, which is similar to the experiment conducted by Wester [10], created temporal correlation artifacts as explained by Porbadnigk *et al* [9]. Thus, according to Porbadnigk *et al* [9], the algorithm seems to classify these temporal effects rather than the EEG signals.

To further verify the temporal correlation effects explained in [9], we invited subject S8 to reconstruct an experiment based on the protocol described in [22]. Concretely, the subject was told to imagine one of the three vowels, /a/, /i/, or /u/ 55 times consecutively in one section. Each section had five blocks, each block started with four beep sounds with 1 s interval between, and then the subject continued to imagine the vowel 11 times at the same rhythm. There was a 2 s break between two consecutive blocks, and a 90 s break between two sections. Thus, each block had 11 s of mental imagery, and we extracted 2 s epochs with 1 s overlap, which yielded 10 epochs for each block. Hence, we obtained total 50 trials for each vowel. We applied our proposed algorithms, which include a MultiCSP preprocessing and Tangent Vector features with RVM or ELM, on this dataset. The accuracy was $(99.3 + / - 1.4)\%$ with min 96.7% and max 100%, using 10 cross validations and 20% evaluation, i.e. 10 testing samples for each class. It can be seen that the accuracy applied on this dataset is significantly higher, which proves that this experiment protocol is inappropriate, and the high recognition rate is erroneously optimistic.

Using HHT to adaptively decompose the signal into the frequency domain, which has been used by Deng *et al* [16] to classify speech imagery, could potentially improve the accuracy of our proposed method. However, HHT is very computationally expensive, especially when we need to process a large number of multiple channels in parallel to align the frequency bands of the decomposed intrinsic mode functions (IMFs) across channels. Furthermore, one also needs to tune the noise-assisted channels to reduce the effects of noise to the IMF [47]. In contrast, our proposed method utilizes Morlet wavelet transform, which is much more computationally efficient and also provide good time-frequency resolution.

In comparison with the mentioned approaches, our proposed method, which is based on the covariance matrix descriptor to fuse the low level features, is not only easier to apply but also improves the discriminative power of the feature. The RVM is found to be more effective than the ELM in our study, as RVM is more robust to noise and outliers thanks to its Bayesian learning models. Finally, as shown in table 3 and the paired t-test scores shown in table 4, our proposed method provides in most cases statistically and significantly better performance than the mentioned approaches found in the literature. The statistical test performed was a two-tailed paired t-test where it was assumed that the variances of the compared quantities were equal.

Table 4. Paired t-test score of the proposed methods and other methods at the 5% significance level. $t(n)$ is the t statistic using n degrees of freedom and p is the p-value.

(a) Vowels

Proposed	Log + LDA	Dasalla *et al* [14]	Min *et al* [45]
Tangent + RVM	$t(7) = 8.56$ $p = 6 \times 10^{-5}$	$t(7) = 9.96$ $p = 2 \times 10^{-5}$	$t(7) = 8.67$ $p = 5 \times 10^{-5}$
Tangent + ELM	$t(7) = 10.09$ $p = 2 \times 10^{-5}$	$t(7) = 9.38$ $p = 3 \times 10^{-5}$	$t(7) = 4.77$ $p = 0.002$

(b) Short words

Proposed	Log + LDA	Dasalla *et al* [14]	Min *et al* [45]
Tangent + RVM	$t(5) = 4.67$ $p = 0.055$	$t(5) = 6.86$ $p = 0.001$	$t(5) = 2.71$ $p = 0.042$
Tangent + ELM	$t(5) = 3.36$ $p = 0.020$	$t(5) = 5.14$ $p = 0.004$	$t(5) = 2.00$ $p = 0.102$

(c) Short versus long words

Proposed	Log + LDA	Dasalla *et al* [14]	Min *et al* [45]
Tangent + RVM (method 1)	$t(5) = 2.09$ $p = 0.091$	$t(5) = 2.79$ $p = 0.038$	$t(5) = 4.64$ $p = 0.006$
Tangent + RVM (method 2)	$t(5) = 3.62$ $p = 0.015$	$t(5) = 4.28$ $p = 0.008$	$t(5) = 5.82$ $p = 0.002$
Tangent + ELM	$t(5) = 2.41$ $p = 0.061$	$t(5) = 4.59$ $p = 0.006$	$t(5) = 6.83$ $p = 0.001$

6.2. Advantages of Riemannian treatment

The Riemannian feature is a high level feature that has been used successfully in many applications, especially in computer vision, such as object recognition, object tracking and action recognition [48–50]. Recently, Riemannian feature has been adopted to the BCI concept, as in the works conducted by Barachant *et al* [32, 51, 52], Congedo *et al* [53], Yger *et al* [54]. As shown in section 6.1, the Riemannian feature is more discriminative comparing with the other features. There are two main advantages of Riemannian treatment.

First, Riemannian features provides a natural way to classify multi-mental tasks, while the conventional CSP is restricted to binary classes in general. By treating covariance matrix as a point on Riemannian space, we can utilize Riemannian geometric distance to classify the data points. There are several distances to discriminate points on Riemannian space, such as log-Euclidean distance, Kullback–Leibler distance, Stein divergence, Von Neumann divergence [28, 55, 56]. In BCI contest, the spatial covariance matrix of the channels has been used to compute CSP. This spatial feature is effective for motor imagery, since right-left hands are controlled by different cortex areas. As pointed by Samek *et al* [57], the CSP and its invariants can be casted to a unifying framework based on Kullback–Leibler divergence between the class covariance matrices. However, the formulation of the classical CSP is mainly based on Reyleigh coefficient between the covariance matrices $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2$ of two classes, i.e.

$$\max_{\mathbf{w}} \frac{\mathbf{w}^{\mathrm{T}}(\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2)\mathbf{w}}{\mathbf{w}^{\mathrm{T}}(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2)\mathbf{w}} \tag{8}$$

which make it inefficient to extend to multiple classes. In contrast, by considering a covariance matrix as a point on Riemannian manifold, any dimension reduction techniques and classifiers, such as LDA or local preserve projection, can be applied by replacing the conventional Euclidean distance with Riemannian distance. Thus, the Riemannian approach generalizes and avoids the restriction to binary classes. For instance, the CSP incorporating *within section stationary regularization (divCSP-WS)* proposed by Samek *et al* [57] is formulated as

$$\begin{aligned}\max_{\boldsymbol{V}} \mathcal{L}(\boldsymbol{V}) =& D_{kl}(\boldsymbol{V}^{\mathrm{T}}\mathbf{\Sigma}_1\boldsymbol{V}||\boldsymbol{V}^{\mathrm{T}}\mathbf{\Sigma}_2\boldsymbol{V}) \\ &- \lambda\frac{1}{N_1+N_2}\sum_{c=1}^{2}\sum_{i=1}^{N_c} D_{kl}(\boldsymbol{V}^{\mathrm{T}}\mathbf{\Sigma}_c^i\boldsymbol{V}||\boldsymbol{V}^{\mathrm{T}}\mathbf{\Sigma}_c\boldsymbol{V})\end{aligned}$$

which is equivalent to LDA in Riemannian space using Kullback–Leibler divergence. In this paper, we utilize the log-Euclidean distance, which essentially approximates the true geometric distance on Riemannian Space by the Euclidean distance between their tangent vectors. This approach is more computationally efficient, and can be utilized directly by any well-established classifiers, i.e. the relevance vector machine with Gaussian kernel in our approach. Nevertheless, other distance, such as Kullback–Leibler or Stein divergence, can also be used. However, the optimization process are more complex due to the nonlinearity of Riemannian distance and computationally expensive [28].

The second advantage is that covariance matrix provides an effective way to fuse heterogeneous features together. Let $X = [x(0), \ldots, x(t), \ldots x(T)] \in \mathbb{R}^{n\times T}$ are the feature vectors collected from T data point samples, assuming zero mean. The covariance matrix can be simply obtained by $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}/T$, regardless of the unit or range of each element in $x(t)$. If $x(t)$ is just the raw signals collected from channels, then applying LDA on Riemannian space using Kullback–Leibler divergence is equivalent to the regularized CSP. However, if mid-level features are extracted from the channels, such as frequency feature and the channel index in our proposed approach, we can incorporate both spatial frequency to the high level feature. In our paper, when classifying short and long words, combining wavelet feature with spatial information has improved the results relatively to using the spatial covariance matrix alone.

The Riemannian feature however still has several limitations. First, it is more computationally expensive than the conventional Euclidean feature vector. Since the distances on Riemannian manifold are computed through eigenvalue decomposition, the dimension of the matrix is preferred to be small. Second, the covariance matrix may be singular or semi-positive definite if the number of samples is small. To overcome these limitations, it is preferable to apply feature selection before computing covariance matrix. In our implementation, this process is obtained by applying CSP.

6.3. Effects of meaning, sounds and complexity to classification accuracy

Based on the presented results, we observed that classification performance between three short words and three vowels is very similar, which suggests that we classified speech imagery based on the sound rather than the meaning.

Second, comparing the performance between classifying three short words and two long words, based on the κ values, indicates that the classification of long words provides better results. Thus, words with higher complexity can be more easily discriminated using EEG signals.

This is also supported by the fact that classification between one short and one long word also yielded the highest κ value. This further suggests that different complexity of the words add an extra degree to discriminate speech imagery.

Although speech imagery has been studied in recent years, to the best of our understanding, this is the first report examining different conditions affecting the speech imagery classification.

7. Conclusion

In this paper, we proposed a novel method to classify speech imagery, and investigated different conditions affecting classification performance. The proposed method is based on the covariance matrix descriptor, in which the low level features extracted from EEG signals are fused to provide more discriminative high level features. The RVM classifier is adopted due to the fact that its Bayesian learning principle provides some important advantages required for BCI applications, such as robustness to large number of outliers and sparse representation. Comparison with other approaches from the literature proved that our method yields significantly better results in term of accuracy and robustness. Intensive study of different conditions related to speech imagery reveals that the sound and complexity of the imagined words are the main mechanisms behind the success of classifying speech imagery using EEG signals. In the future works, we would like to combine speech imagery with other modalities, such as motor and visual imagery to provide more degrees of freedom for BCI applications.

Acknowledgment

This work is supported by the US Defense Advanced Research Projects Agency (DARPA) grant D14AP00068 and US Air Force Office of Scientific Research (AFOSR) award FA9550-14-1-0149.

References

[1] Tanaka K, Matsunaga K and Wang H O 2005 Electroencephalogram-based control of an electric wheelchair *IEEE Trans. Robot.* **21** 762–6
[2] Rebsamen B, Teo C L, Zeng Q, Ang M H, Burdet E, Guan C, Zhang H and Laugier C 2007 Controlling a wheelchair indoors using thought *IEEE Intell. Syst.* **22** 18–24
[3] Mondada L, Karim M E and Mondada F 2016 Electroencephalography as implicit communication channel for proximal interaction between humans and robot swarms *Swarm Intell.* **10** 1–19
[4] Esfahani E T and Sundararajan V 2012 Classification of primitive shapes using brain-computer interfaces *CAD Comput. Aided Des.* **44** 1011–9
[5] Contreras S and Sundararajan V 2016 Visual imagery classification using shapelets of EEG signals *ASME 2012 Int. Design Engineering Technical Conf. and Computers and Information in Engineering Conf.* pp 1–6
[6] Herff C and Schultz T 2016 Automatic speech recognition from neural signals: a focused review *Frontiers Neurosci.* **10** 1–7
[7] Denby B, Schultz T, Honda K, Hueber T, Gilbert J M and Brumberg J S 2010 Silent speech interfaces *Speech Commun.* **52** 270–87
[8] Schultz T 2010 ICCHP keynote: recognizing silent and weak speech based on electromyography *Lect. Not. Comput. Sci.* **6179** 595–604 (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
[9] Porbadnigk A, Wester M, Calliess J-P and Schultz T 2009 EEG-based speech recognition impact of temporal effects *Int. Conf. on Bio-inspired Systems and Signal Processing*
[10] Wester M and Schultz T 2006 Unspoken speech-speech recognition based on electroencephalography *PhD Thesis* Universität Karlsruhe (TH), Karlsruhe, Germany
[11] D'Zmura M, Deng S, Lappas T, Thorpe S and Srinivasan R 2009 Toward EEG sensing of imagined speech *Int. Conf. on Human-Computer Interaction* pp 40–8
[12] Kim J, Lee S-K and Lee B 2014 EEG classification in a single-trial basis for vowel speech perception using multivariate empirical mode decomposition *J. Neural Eng.* **11** 036010
[13] Brigham K and Kumar B V K V 2010 Imagined speech classification with EEG signals for silent communication: a preliminary investigation into synthetic telepathy *4th Int. Conf. on Bioinformatics and Biomedical Engineering* pp 1–4
[14] DaSalla C S, Kambara H, Sato M and Koike Y 2009 Single-trial classification of vowel speech imagery using common spatial patterns *Neural Net.* **22** 1334–9
[15] DaSalla C S C, Kambara H, Koike Y and Sato M 2009 Spatial filtering and single-trial classification of EEG during vowel speech imagery *Int. Convention rehabilitation Engineering and Assistive Technology* vol 5 p 27
[16] Deng S, Srinivasan R, Lappas T and D'Zmura M 2010 EEG classification of imagined syllable rhythm using Hilbert spectrum methods *J. Neural Eng.* **7** 046006
[17] Idrees B M and Farooq O 2016 Vowel classification using wavelet decomposition during speech imagery *3rd Int. Conf. on Signal Processing and Integrated Networks* pp 636–40
[18] Chi X, Hagedorn J B, Schoonover D and Zmura M D 2011 EEG-Based discrimination of imagined speech phonemes *Int. J. Bioelectromagn.* **13** 201–6
[19] Riaz A, Akhtar S, Iftikhar S, Khan A A and Salman A 2015 Inter comparison of classification techniques for vowel speech imagery using EEG sensors *2nd Int. Conf. on Systems and Informatics* pp 712–7
[20] Suppes P, Lu Z L and Han B 1997 Brain wave recognition of words *Proc. Natl Acad. Sci. USA* **94** 14965–9
[21] Suppes P, Han B and Lu Z-L 1998 Brain-wave recognition of sentences *Proc. Natl Acad. Sci.* **95** 15861–6
[22] González-Castañeda E F, Torres-García A A, Reyes-García C A and Villaseñor-Pineda L 2017 Sonification and textification: proposing methods for classifying unspoken words from EEG signals *Biomed. Signal Process. Control* **37** 82–91
[23] Salama M, Elsherif L, Lashin H and Gamal T 2014 Recognition of Unspoken Words Using Electrode Electroencephalograhic Signals *The 6th Int. Conf. on Advanced Cognitive Technologies and Applications* pp 51–5

[24] Wang L, Zhang X, Zhong X and Zhang Y 2013 Analysis and classification of speech imagery EEG for BCI *Biomed. Signal Process. Control* **8** 901–8
[25] Mohanchandra K and Saha S 2016 A communication paradigm using subvocalized speech: translating brain signals into speech *Augmented Human Res.* **1** 3
[26] Tu L W 2010 *An Introduction to Manifolds* (Berlin: Springer)
[27] Shawe-Taylor J and Cristianini N 2004 *Kernel Methods for Pattern analysis* (Cambridge: Cambridge university press)
[28] Nguyen C H and Artemiadis P 2017 EEG feature descriptors and discriminant analysis under Riemannian manifold perspective *J. Neurocomput.* (https://doi.org/10.1016/j.neucom.2017.10.013)
[29] Damoulas T and Girolami M A 2008 Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection *Bioinformatics* **24** 1264–70
[30] Psorakis I, Damoulas T and Girolami M A 2010 Multiclass relevance vector machines : sparsity and accuracy *IEEE Trans. Neural Netw.* **21** 1588–98
[31] Bishop C M 2006 *Pattern Recognition and Machine Learning* (Berlin: Springer)
[32] Barachant A, Bonnet S, Congedo M and Jutten C 2012 Multiclass brain-computer interface classification by Riemannian geometry *IEEE Trans. Biomed. Eng.* **59** 920–8
[33] Klem G, Luders H, Jasper H and Elger C 1958 The ten-twenty electrode system of the International Federation *Electroencephalogr. Clin. Neurophysiol.* **10** 371–5
[34] He P, Wilson G and Russell C 2004 Removal of ocular artifacts from electro-encephalogram by adaptive filtering *Med. Biol. Eng. Comput.* **42** 407–12
[35] Koles Z J, Lazar M S and Zhou S Z 1990 Spatial patterns underlying population differences in the background EEG *Brain Topogr.* **2** 275–84
[36] Blankertz B, Tomioka R, Lemm S, Kawanabe M and Müller K-R 2008 Optimizing spatial filters for robust EEG single-trial analysis *IEEE Signal Process. Mag.* **25** 41–56
[37] Dornhege G, Blankertz B, Curio G and Muller K 2004 Increase information transfer rates in BCI by CSP extension to multi-class *Adv. Neural Inf. Process. Syst.* **16** 733–40
[38] Lotte F, Guan C, Lotte F, Guan C and Member S 2010 Regularizing common spatial patterns to improve BCI designs: theory and algorithms *IEEE Trans. Biomed. Eng.* **58** 355–62
[39] Grosse-Wentrup M and Buss M 2008 Multiclass common spatial patterns and information theoretic feature extraction *IEEE Trans. Biomed. Eng.* **55** 1991–2000
[40] Martin S, Brunner P, Holdgraf C, Heinze H-J, Crone N E, Rieger J, Schalk G, Knight R T and Pasley B N 2014 Decoding spectrotemporal features of overt and covert speech from the human cortex *Frontiers Neuroeng.* **7** 14
[41] Martin S, Brunner P, Iturrate I, Millán J D R, Schalk G, Knight R T and Pasley B N 2016 Word pair classification during imagined speech using direct brain recordings *Sci. Rep.* **6** 25803
[42] Tomioka R and Muller K R 2010 A regularized discriminative framework for EEG analysis with application to brain-computer interface *NeuroImage* **49** 415–32
[43] Cichocki A, Amari S I, Siwek K, Tanaka T, Phan A H, Zdunek R and Bakardjian H 2007 ICALAB toolboxes for signal processing (www.bsp.brain.riken.jp/ICALAB)
[44] Neuper C and Pfurtscheller G 2010 *Neurofeedback Training for BCI Control* (Berlin: Springer) pp 65–78
[45] Min B, Kim J, Park H-j and Lee B 2016 Vowel imagery decoding toward silent speech BCI using extreme learning machine with electroencephalogram *BioMed. Res. Int.* **2016** 2618265
[46] Huang G-B, Zhou H, Ding X and Zhang R 2012 Extreme learning machine for regression and multiclass classification *IEEE Trans. Syst. Man Cybern. Part B, Cybern.* **42** 513–29
[47] Mandic D P 2011 Filter bank property of multivariate empirical mode decomposition *IEEE Trans. Signal Process.* **59** 2421–6
[48] Harandi M T, Hartley R, Lovell B and Sanderson C 2016 Sparse coding on symmetric positive definite manifolds using Bregman divergences *IEEE Trans. Neural Netw. Learn. Syst.* **27** 1294–306
[49] Cherian A and Sra S 2016 Riemannian dictionary learning and sparse coding for positive definite matrices *IEEE Trans. Neural Netw. Learn. Syst.* **28** 2859–71
[50] Palhang M, Faraki M and Sanderson C 2015 Log-Euclidean bag of words for human action recognition *IET Comput. Vis.* **9** 331–9
[51] Barachant A, Bonnet S, Congedo M and Jutten C 2010 Common spatial pattern revisited by Riemannian geometry *IEEE Int. Workshop on Multimedia Signal Processing* pp 472–6
[52] Barachant A, Bonnet S, Congedo M and Jutten C 2013 Classification of covariance matrices using a Riemannian-based kernel for BCI applications *Neurocomputing* **112** 172–8
[53] Congedo M, Barachant A and Andreev A 2013 A new generation of brain–computer interface based on Riemannian geometry vol 33 (arXiv:1310.8115)
[54] Yger F, Berar M and Lotte F 2016 Riemannian approaches in brain-computer interfaces: a review *IEEE Trans. Neural Syst. Rehabil. Eng.* **4320** 1–10
[55] Harandi M, Salzmann M and Hartley R 2017 Dimensionality reduction on SPD manifolds: the emergence of geometry-aware methods *IEEE Trans. Pattern Anal. Mach. Int.* (https://doi.org/10.1109/TPAMI.2017.2655048)
[56] Sra S 2012 A new metric on the manifold of kernel matrices with application to matrix geometric means *Adv. Neural Inf. Process. Syst.* **25** 144–52
[57] Samek W, Kawanabe M and Muller K R 2014 Divergence-based framework for common spatial patterns algorithms *IEEE Rev. Biomed. Eng.* **7** 50–72