Comparison of Generic and Subject-Specific Training for Features Classification in P300 Speller

Ayana Mussabayeva, Prashant Kumar Jamwal, and Muhammad Tahir Akhtar
School of Engineering and Digital Sciences
Nazarbayev University, Nur-Sultan, Kazakhstan
E-mails: {ayana.mussabayeva, prashant.jamwal, muhammad.akhtar}@nu.edu.kz, akhtar@ieee.org

***Abstract*—Designing subject-adaptive brain-computer interface (BCI) systems using event-related potential (ERP) paradigm is a challenging problem for BCI researchers, as ERP response to the same visual stimuli varies from one human to another. In this paper two different training approaches, subject-specific training (SST) and generic training (GT), are proposed. The first approach employs training classifiers for each subject separately, while the second approach shuffles all the data from different subjects and train classification model on merged data. The proposed approaches are tested for three features classification algorithms: support vector machine (SVM), k-nearest neighbours (kNN) and linear discriminant analysis (LDA). It has been found that the proposed GT approach is very efficient for training kNN classifier, reaching averagely 98% accuracy, while it does not have any noticeable improvements when using LDA. SVM classifier turned to be non-efficient for classification of the target ERP component while using both training approaches.**

***Index Terms*—Brain-computer interface, P300 Speller, support vector machine, k-nearest neighbours, linear discriminant analysis, EEG features classification.**

I. Introduction

Brain-Computer Interface (BCI) can be defined as closed-loop system which processes human-brain signals for enabling the interaction with the computer-based device. Generally, a typical BCI system acquires brain signals, processes them and translates those signals into the device commands [1]. There is a number of applications of BCI, such as assisting devices for disabled people or diagnostic medical devices. Apart from the medical devices, recently BCI systems have started occupying entertainment industry, promising engaging brain-controlled games for users [2].

Data acquisition in BCI systems can be performed by using various neuroimaging techniques, which detect either the hemodynamic response (electro-chemical reactions between the neurons) of the human brain or the electrophysiological activity (ionic currents generated by the neurons). The most useful technique for reading brain activity in BCI systems is electroencephalography (EEG), as it is non-invasive and portable. The possible disadvantage of EEG is its low spatial resolution, which much worse than the spatial resolution of such methods as magnetoencephalography (MEG), near-infrared spectroscopy (NIRS) and functional magnetic resonance imaging (fMRI). Nevertheless, temporal resolution of EEG is only about 1 ms, while it takes 1-10 s to receive the brain response for other neuroimaging techniques [3].

P300 Speller is a classical EEG-based BCI system proposed in late 1980s by L.A. Farwell and E. Donchin [4]. P300 Speller is designed to enable people with spelling difficulties and various diseases as amyotrophic lateral sclerosis (ALS) to communicate with the outer world. There are many different methods being used for constructing the BCI speller systems for facing various user needs, for example user's visual capability. Classical matrix speller (which has an interface of a matrix of characters with flashing rows and columns) requires its user to be able to move their eyes and fix their sight on a particular item of the character matrix. There are different types of interfaces designed for users with poor visual capability. In rapid serial visual presentation (RSVP) Speller, where the characters are shown one-by-one in the center of the graphical user interface (GUI) of the system [5]. Auditory-based speller systems, such as auditory multi-class spatial ERP (AMUSE) speller, provide audio interface for the users instead of GUI [6].

In this paper, the classical 6×6 matrix speller using event-related potential (ERP) paradigm is considered [4]. The ERP-based spellers are always called P300 Spellers as they are using the positive voltage peak of the brainwave occurring approximately 300 ms after the target stimuli. If the chosen character flashes in the matrix, there will be a significant voltage peak noted 300 ms after it flashes. However, the latency of P300 ERP component can vary from 250 ms to 500 ms from user to user.

Three algorithms for EEG signal features classification are reviewed and compared. Support vector machine (SVM), linear discriminant analysis (LDA) and k-nearest-neighbours (kNN) classifiers with different parameters (kernel functions, solvers and number of neighbours) are trained and tested using different training approaches. Training approaches proposed are subject-specific training (SST) and generic training (GT). In SST, each subject data is used separately for training the model. In GT, the classifier is trained on a merged data from different subjects.

SVM is frequently used for brain signal features classification and sometimes provides good results. For example, the average accuracy of SVM binary classifier in emotion recognition task (88.5%) was significantly higher than the average accuracy of the Naive Bayes algorithm (60.2 %) and kNN classifier (71.8 %) [7]. SVM also outperformed LDA in discriminating early vascular dementia patients [8]. LDA is also a popular choice for features classification in BCI systems, e.g. when using EEG and electrooculography (EOG) combined together for detecting user's response, LDA achieves accuracy of 97.6% [9]. kNN algorithm is not as popular in BCI research as SVM and LDA, however, despite the simplicity of the given classifier, it can outperform the accuracy of some other classification algorithms such as naive bayesian (NB) or voting-extreme learning machine (V-ELM) [7].

The remainder of this paper is organized as follows: Section II reviews the dataset used for the experiments and introduces the classifiers (SVM, kNN and LDA) and their performance evaluation techniques used. Section III represents the results obtained while training the classifiers using SST and GT approaches. Finally, the conclusions are represented in Section IV.

II. Methodology

A. Dataset

The dataset used in this project is Akimpech P300 dataset [10], which contains EEG recordings of 20 healthy subjects aged from 21 to 25 years participating in P300 Speller training. EEG signal provided by the dataset had been preprocessed with notch filter (Chebyshev 4th order filter with frequency range 58-62 Hz) and band-passed with Chebyshev 8th order filter with the frequency range 0.1-60 Hz. The dataset contains three main variables. The most important variable is X, containing the EEG values for each row/column flashing trial.

$$X = [x_1, x_2, ...x_{140}], \tag{1}$$

X is a brainwave response to the target stimuli which has 140 values. All the datapoints in the row combined together give us a voltage peak, which can be analyzed and classified further as a target or non-target ERP peak.

Y is the classification label, which is either 1 or -1 for target and non-target response, respectively, i.e.

$$y \in \{1, -1\}. \tag{2}$$

In order to identify which character is chosen by the subject, c variable is introduced. It represents the number of the flashing row or column in the matrix of characters as follows:

$$c \in \{1, 2, 3, ..., 12\}. \tag{3}$$

The Fig.1 represents the assigned c values to rows and columns of P300 Speller matrix.

B. Classifiers

In this paper, there are three main types of classification algorithms used for classification of brain response on visual stimuli in P300 Speller: SVM, kNN and LDA. SVM is a supervised learning algorithm, which can be efficiently used for regression and classification problems. In this paper, SVM binary classifier is considered for identifying target and non-target brain responses. SVM binary classifier constructs the optimal hyperplane in order to separate two classes of the data, maximizing the margin of separation. Data separation is performed in two steps, which are kernel trick and quadratic optimization problem.

Kernel trick is a transformation of the input data into a high-dimensional feature space using kernel functions. Gaussian radial basis function (RBF) is a general-purposed kernel function of SVM classifier. It is usually used when there is no prior knowledge about the input data. It takes feature vectors x_i, x_j and uses standard deviation σ, and is given as:

$$K(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right). \tag{4}$$

Another popular kernel function of SVM is a polynomial kernel, given as:

$$K(x_i, x_j) = (x_i x_j + 1)^d, \tag{5}$$

where d denotes the degree of the polynomial function and it may vary depending on the input data.

The third type of kernel considered in this paper is a sigmoid function, calculated as:

$$K(x_i, x_j) = \frac{1}{1 + \exp\left(-k x_i x_j\right)}, \tag{6}$$

where the coefficients k and b can also be tuned depending on the input data.

After applying kernel trick, the optimal hyperplane is found by solving quadratic optimization problem. For solving this problem usually derivative tests are applied, such as Karush-Kuhn-Tucker (KKT) conditions. KKT conditions can be applied for solving quadratic optimization problem in adaptive SVM, when it is necessary to implement continuous learning of SVM classifier by adding new training data to the existing solution. Such approach has been used for incremental SVM (ISVM) in [11].

ISVM with linear kernel can also be used for implementing Dynamic Stopping (DS) continuous learning approach. DS can be applied in P300 Spellers for reducing the training time of the system for each particular user. When applying DS approach, the classifier is trained on the data until some stopping condition is met. Sometimes stopping conditions are formulated by introducing stopping parameters θ_i and stopping conditions $\theta_i \geq \hat{\theta}_i$. When the stopping condition is satisfied, the system specifies the target ERP peak and the chosen character accurately enough, so it is not necessary to continue learning iterations [12].

Another classifier used in this paper is kNN classifier, which calculates the distance (Euclidian distance, cosine distance and etc.) between the classified data point and its k neighbours. This type of classifier is simple and fast in terms of running time complexity. In this paper, the correlation between the number of neighbours k and the classifier's performance is mainly concerned. The number of neighbours k should be odd number, as in our case the binary classifier is considered. The distance metric used is Minowski distance with the power of $p = 2$, resulting a simple normed Euclidean distance. As

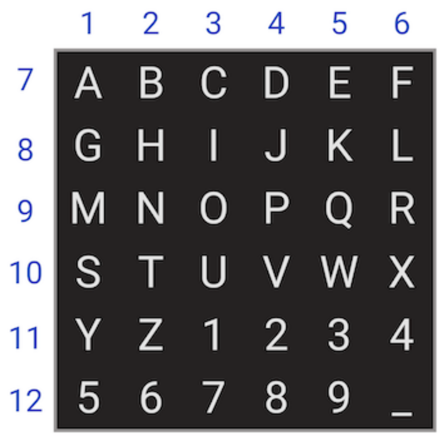

Fig. 1. Labels of numbered rows/columns of P300 Speller 6×6 matrix stored in c variable

we use single averaged EEG signal, for now it is enough to compute distance between the values as between two data points in a plane and use Euclidean metric.

Apart from that, the performance of LDA for binary classification of brain response is overviewed. LDA classifier is a simple algorithm, that maximizes the distance between two mean classes and minimizes the variance between two classes: target and non-target brain response.

One of the main advantage of LDA classifier is that it can be used for continuous learning without complex modifications, as it enables adding new data to the existing equation. LDA classifier is frequently used for continuous learning of P300 Speller with Bayesian probabilistic approach of DS [13]. LDA with least squares solution (LSS), eigenvalue decomposition (ED) and singular value decomposition (SVD) solvers is used for performance evaluation in this paper. LSS and ED solvers are also combined with shrinkage using Ledoit-Wolf lemma [14].

C. Performance Evaluation

In order to evaluate the performance of each classifier, the number of true positive (TP), true negative (TN), false positive (FP) and false negative (FN) predictions are calculated. For performance assessment, K-fold cross validation technique is used for training and testing the classification models. In this/ paper, the number of folds used is $K = 10$, which means that the data is split into ten folds, nine of which are used as training data and one for testing.

The accuracy (A) for each epoch is calculated as:

$$A = \frac{TP + TN}{TP + TN + FP + FN}. \tag{7}$$

In order to check, whether the target class is correctly recognized and the number of FN is low, recall (R) is calculated as:

$$R = \frac{TP}{TP + FN}. \tag{8}$$

Precision value (P) indicates an EEG signal labelled as positive (target response) is positive indeed and is computed as:

$$P = \frac{TP}{TP + FP}. \tag{9}$$

The EEG signals from 20 subjects ($n = 20$) are used for training and testing the classifiers. There are two approaches proposed in this paper. The first method is SST, which means training classifiers for each subject separately and presenting the average performance metrics, e.g. the accuracy is calculated as:

$$\bar{A} = \frac{1}{n}\sum_{i=1}^{n} A_i. \tag{10}$$

The second approach proposed is GT, which is training the classifiers independently from subjects. We assume that the classifiers trained on a dataset consisted from merged data from 20 subject may have better performance than when training classifiers for each subject separately. In order to verify classifiers trained on the merged dataset, the number K in K-fold validation can be increased, as the size of dataset is increased 20 times, so in GT 200-fold validation is used for each analyzed classifier.

III. Results and Discussion

A. Subject-Specific Training (SST)

For SVM Classifier there have been four polynomial kernel, Guassian RBF kernel and sigmoid kernel classifiers trained and tested. The results, presented in the Table I, show that the accuracy of Guassian RBF SVM is the highest, however the

TABLE I
SUBJECT-SPECIFIC TRAINING OF SVM, KNN AND LDA

Classifier	Parameters	Accuracy (%)	Recall(%)	Precision(%)
SVM	**Kernel type**			
	Gaussian RBF	83.33	0.24	0.03
	Sigmoid	78.33	11.56	21.72
	Polynomial, d=2	81.26	0.55	0.52
	Polynomial, d=3	82.14	3.85	7.56
	Polynomial, d=4	80.43	4.69	17.45
	Polynomial, d=5	81.35	5.42	23.49
kNN	**k (number of neighbours)**			
	3	77.55	10.41	21.43
	5	81.1	5.41	25.95
	7	81.8	2.08	16.1
	9	82.21	1.25	14.66
	11	82.9	1.66	23.99
	13	83.08	1.46	26.66
LDA	**Solver**			
	LSS	88	48.33	71.72
	LSS, with shrinkage	84.17	11.18	80.78
	ED	88	48.33	71.72
	ED, with shrinkage	84.17	11.18	80.78
	SVD	88	48.33	71.72

recall and precision are too low, meaning that the classifier fails to predict target response. At first sight SVM provides good accuracy, however low recall and precision show that it fails on imbalanced data.

By analyzing the results obtained, it can be said that SVM with sigmoid kernel function has the best performance, as its recall and precision are higher than when using other kernels. However, the accuracy of this classifier is only 78.33%.

SST of kNN classifier with different number of neighbours performs slightly better than SVM. It is seen from the Table I that kNN with low k value works better on imbalanced data providing approximately 10% of recall and 20% of precision.

LDA classifier for separate training shows the highest accuracy of 88%, but low recall and precision values. It is seen from the Table I that using different solvers (LSS, ED or SVD) affects only the run-time complexity and does not impact the other performance parameters. General loss in performance can be noticed when using automatic shrinkage parameter calculated by Ledoit-Wolf lemma. Thus, it can be summarized that it is not necessary to use shrinkage estimator as the dataset is large enough.

B. Generic Training (GT)

The results from Table I show that it does not matter which solver is used for LDA in our case, thus, for GT experiment, only SVD LDA solver is used.

When applying GT for all of the 20 subjects data, the data is shuffled and classifiers are verified using K-fold cross-validation. The merged dataset is more subject-independent, combining folds consisting of data from different subjects.

The results presented in the Table II show the performance evaluation of all classifiers discussed above using GT approach. SVM takes much more time to train and test the classification model, while kNN and LDA classifiers have an advantage of fast data processing. SVM training requires approximately 18 hours while it takes less than 20 minutes to train and test LDA and kNN classifiers. This can be explained by the run-time complexity of the kernel functions in SVM classifiers and quadratic optimization problem solving time. Nevertheless, some improvement of the performance can be noticed for SVM when using GT training approach. Sigmoid kernel remains the best choice for both SST and GT approaches. Still, SVM turned out to be the most inefficient classifier among the reviewed ones, as its recall and precision values are too small and the running time is too long. This can be explained by the fact that SVM fails to work properly on imbalanced datasets and for achieving better results in P300 classification using SVM it is necessary to perform some data pre-processing for balancing the data. Anyway, it can be seen that SVM classifier is also inefficient in terms of time complexity.

LDA classifier does not seem to be improved by GT approach, as its accuracy decreased by 1.34% compared with SST trained classifier. LDA classifier models the difference between the classes of data by minimizing the within-class variance and maximizing the between-class variance. It can be noticed that when trained on a merged dataset the within-class variance is larger than in SST approach. That is why the performance slightly decreases. Nevertheless, it can be concluded that LDA classifier provides relatively stable results for both training approaches.

The most significant improvement can be noticed when

TABLE II
GENERIC TRAINING OF SVM, KNN AND LDA

Classifier	Parameters	Accuracy (%)	Recall(%)	Precision(%)	Running time
	Kernel				**(hours)**
	Gaussian RBF	89.49	4.03	8.35	17.07
	Sigmoid	73.06	15.56	25.63	18.31
SVM	Polynomial, d=2	83.86	2.33	14.52	17.04
	Polynomial, d=3	85.65	6.72	8.91	18.67
	Polynomial, d=4	85.62	6.83	20.05	18.95
	Polynomial, d=5	85.62	8.82	25.58	19.12
	k (number of neighbours)				**(minutes)**
	3	99.39	97.83	98.3	4.05
	5	98.98	96.38	97.01	5.36
kNN	7	97.62	91.62	93.94	7.69
	9	98.28	93.68	95.14	8.58
	11	97.42	90.54	92.53	10.1
	13	97.13	89.41	91.65	11.54
LDA	**Solver**				**(minutes)**
	SVD	86.66	35.3	70.79	7.87

training kNN on a merged data, which gives us an average accuracy of 98.13% with the recall value of 93.24% and 94.76% average precision. While the average accuracy during SST was 77.08% with the recall value of only 13.2% and precision value of 30%. It is seen that the most efficient classifier uses GT with the minimal number of neighbours ($k = 3$) and achieves 99.39% accuracy. The running time can be explained by the fact that kNN is an instance based learning algorithm, which stores the training data and compares the test data with the training set only during the testing process. The time required for training is very low, while for testing it increases in GT approach, however the classifier still works really fast compared to other models, meaning that kNN is fine with processing averaged EEG signal data. On the other hand it may be assumed that the testing time for multi-channel EEG using kNN will increase significantly compared to LDA. The advantage of kNN is that the data can be added seamlessly, which evokes opportunities for building subject-independent P300 Speller. The versatility of the merged dataset provides better accuracy and does not effect significantly on the testing time, thus it can be concluded that kNN is a good choice for P300 classification in averaged EEG channel.

C. A Few Remarks

One of the limitations of this study is that it does not consider multi-channel EEG signal. However, it can be supposed that LDA would outperform SVM the same way when using multi-channel signal, after data dimension reduction using the following techniques:

- Geometry-based data alignment techniques like Euclidian Alignment(EA) or Riemannian Geometry-based classifiers [15].
- Neural networks for features extraction from multi-channel EEG signal, such as convolutional neural network (CNN) [16] and long-short term memory (LSTM) network [17].

The possible application of the employed method:

- The reviewed classifiers can be applied for continuous learning, which is a popular research topic, that faces the problem of the implementation of a stable user-adaptive (or subject-independent) P300 Speller.
- GT training approach can be used for pre-training of the classifier and further addition of the new data to the existing solution.
- LDA and kNN can be used for adaptive training, while SVM requires some complex modifications to make it adaptive and transforming it into ISVM.

IV. CONCLUSIONS

In this paper, we applied two different training approaches (SST and GT) when training SVM, kNN and LDA classifiers on EEG data for P300 Speller. It has been found that GT approach gives us a better performance of classification target brain response when training on an imbalanced dataset. SVM classifiers showed very low recall and precision values, meaning that it still overfits significantly for both SST and GT approaches. Moreover, SVM has a significant loss in running time compared to LDA and kNN classifiers. By comparing the results of kNN classifier when using SST and GT, it can be summarized that kNN classifier provides the best performance when applying GT. LDA, however, is more preferable when training for each subject separately.

In the future the multi-channel EEG features classification is going to be considered using GT and SST training approaches. The following steps are going to be implemented further:

- EEG data collection from healthy subjects using g.USBamp-Research digital amplifier.
- Classification of averaged EEG signal's features using LDA and kNN classifiers.

- Classification of multi-channel EEG data with CNN and LSTM using GT and SST approaches.

ACKNOWLEDGMENT

This work was partially supported by the Faculty Development Competitive Research Grants Program of Nazarbayev University under Grant Number 110119FD4525.

REFERENCES

[1] R. Rao and R. Scherer, "Brain-computer interfacing," *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 152 – 150, Jul. 2010.
[2] S. Amiri, R. Fazel-Rezai, and V. Asadpour, "A review of hybrid brain-computer interface systems," *Adv. Human-Computer Interaction*, vol. 2013, 2013.
[3] L.F. Nicolas-Alonso and J. Gomez-Gil. "Brain computer interfaces, a review," *Sensors*, vol. 12, no. 2, pp. 1211–1279, 2012.
[4] L. Farwell and E. Donchin, "Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials," *Electroencephalogr Clin Neurophysiol*, vol. 70, pp. 510-23, Dec. 1988.
[5] L. Acqualagna, B. Blankertz, "Gaze-independent BCI-spelling using rapid serial visual presentation (RSVP)," *Clinical Neurophysiology*, vol. 124, no. 5, pp. 901-908, 2013.
[6] N. Simon, I. Käthner, C. A. Ruf, E. Pasqualotto, A. Kübler, and S. Halder, "An auditory multiclass brain-computer interface with natural stimuli: Usability evaluation with healthy participants and a motor impaired end user," *Front. Hum. Neurosci.*, vol. 8, no. 1, pp. 1–14, 2015.
[7] H. R. Hou, X. N. Zhang, and Q. H. Meng, "Odor-induced emotion recognition based on average frequency band division of EEG signals," *J. Neurosci. Methods*, vol. 334, art. no. 108599, 2020.
[8] C. Wang, J. Xu, S. Zhao, and W. Lou, "Identification of early vascular dementia patients with EEG signal," *IEEE Access*, vol. 7, pp. 68618–68627, 2019.
[9] M. H. Lee, J. Williamson, D. O. Won, S. Fazli, and S. W. Lee, "A high performance spelling system based on EEG-EOG signals with visual feedback," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 26, no. 7, pp. 1443–1459, Jul. 2018.
[10] O. Yanez-Suarez, L. Bougrain, C. Saavedra, E. Bojorges, and G. Gentiletti. (2011). P300-Speller Public-Domain Database. [Online]. Available: http://akimpech.izt.uam.mx/p300db
[11] C. P. Diehl and G. Cauwenberghs, "SVM incremental learning, adaptation and optimization," in *Proc. Int. Joint Conf. Neural Netw.*, vol. 4, pp. 2685–2690, 2003.
[12] K. Vo, T. Pham, D. N. Nguyen, H. H. Kha, and E. Dutkiewicz, "Subject-independent ERP-based brain–computer interfaces," *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 26, no. 4, pp. 719-728, Apr. 2018.
[13] C. S. Throckmorton, K. A. Colwell, D. B. Ryan, E. W. Sellers, and L. M. Collins, "Bayesian approach to dynamically controlling data collection in P300 Spellers," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 21, no. 3, pp. 508–517, 2013.
[14] O. Ledoit and M. Wolf, "A well-conditioned estimator for large-dimensional covariance matrices," *J. Multivar. Anal.*, vol. 88, no. 2, pp. 365–411, 2004.
[15] He He and Dongrui Wu, "Transfer learning for brain-computer interfaces: Euclidean space data alignment approach," *IEEE Trans. Biomedical Eng.*, vol. 26, no. 2, pp. 399-410, 2020.
[16] Z. Chen and X. Zhang, "Dynamic stopping in P300 Speller with convolutional neural network," in *Proc. Int. IEEE/EMBS Conf. Neural Eng. NER*, pp. 383–386, 2017.
[17] A. Ditthapron, N. Banluesombatkul, S. Ketrat, E. Chuangsuwanich, and T. Wilaipraistporn, "Universal joint feature extraction for P300 EEG classification using multi-task autoencoder," *IEEE Access*, vol. 7, pp. 68415–68428, 2019.