HAL
open science

Independent Component Analysis for EEG Data Preprocessing - Algorithms Comparison

Izabela Rejer, Pawel Górski

To cite this version:

Izabela Rejer, Pawel Górski. Independent Component Analysis for EEG Data Preprocessing - Algorithms Comparison. 12th International Conference on Information Systems and Industrial Management (CISIM), Sep 2013, Krakow, Poland. pp.108-119, 10.1007/978-3-642-40925-7_11 . hal-01496056

HAL Id: hal-01496056

https://inria.hal.science/hal-01496056v1

Submitted on 27 Mar 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Independent Component Analysis for EEG Data Preprocessing - Algorithms Comparison

Izabela Rejer and Paweł Górski

West Pomeranian University of Technology, Szczecin;
Faculty of Computer Science;
Zolnierska 49, 71-210 Szczecin
irejer@wi.zut.edu.pl, pagorski@wi.zut.edu.pl

Abstract. Some scientific papers report that when Independent Component Analysis (ICA) is applied in the preprocessing step of designing a brain computer interface, the quality of this interface increases. At the same time, however, these papers do not provide information about the exact gain in classification precision obtained after applying different ICA algorithms. The aim of this paper is to compare three algorithms for Independent Component Analysis applied in the process of creating a brain computer interface in order to find out whether the choice of a specific ICA algorithm has an influence on the final classification precision of this interface. The comparison will be carried out with a set submitted to the second BCI Competition.

Keywords: Brain Computer Interface, BCI, EEG, preprocessing, ICA, Independent Component Analysis

1 Introduction

A BCI (Brain Computer Interface) is defined as a communication system in which messages or commands that a user sends to the external world do not pass through the brain's normal output pathways of peripheral nerves and muscles [1]. Although, at first brain computer interfaces were dedicated mainly to people in locked-in state or in completely locked-in state, nowadays, the range of their potential recipients is much wider. They are tested and used by the army, they are used in the entertainment industry and there are also some attempts to use them to control limb prosthesis.

A BCI system is composed of seven parts, followed one after the other in a closed-loop. Succeeding parts of the BCI system are responsible for: measuring brain activity, preprocessing of the acquired signals, describing the signals by a few relevant features (feature extraction), selecting the most relevant features (feature selection), assigning a class to a set of selected features (classification), executing a command assigned to the chosen class and providing feedback to the user informing him about the mental state recognized by the BCI system. Although, all steps of the loop have to be carefully designed in order to build a successful BCI system, four of them play the major role: preprocessing [2-4], feature extraction [5-6], feature selection [7-9] and

K. Saeed et al. (Eds.): CISIM 2013. LNCS 8104, p. 99-110, 2013.
© IFIP International Federation for Information Processing 2013

classification [10-12]. This paper deals with the first of these four main steps, it is with the preprocessing step. The aim of this step is to transform the acquired brain signals to its purer form by eliminating artifacts, reducing the ongoing brain activity, enhancing the primary components. One of the methods used in the preprocessing step is Independent Component Analysis (ICA).

Independent Component Analysis is a method for transforming a set of mixed signals into a set of independent components. It has been reported in some publications that this method allows to detect artifacts in originally recorded signals [13]. Due to this, ICA seems to be a good choice for preprocessing of electroencephalographic data (EEG) used as control signals in brain computer interfaces (BCI). After applying ICA on a set of EEG data, some components should reflect original data sources and one or more components should reflect artifacts. In the process of feature selection, which is a further step of signal processing leading to classification, features calculated for the "artifact components" should be discarded from the set of features while features calculated for the components essential for the classification precision should be preserved.

One can ask why features calculated for the "artifact components" will be discarded from the feature set. The answer is quite straightforward. There are two main types of artifacts activity. Some artifacts, such as eye artifact, appear rhythmically during each trial, while others, such as unexpected body movements, appear in unexpected moments along the whole experiment. Since artifacts from both groups do not dependent on the class coded in the recorded signals, they do not enhance the classification precision. Hence, when the classification precision is a measure used in the feature selection process, features calculated for the components reflecting artifacts should be discarded from the feature set.

There are a lot of scientific papers describing applications of ICA in BCI research [13-15]. In most of them there is underlined that by applying ICA for data preprocessing, the classification precision increases. However, these papers do not specify the exact profits (in terms of classification accuracy) gained by using different algorithms for calculating independent components. Does it mean that the choice of the ICA algorithm is of no importance for the classification precision?

The aim of this paper is to examine whether ICA transformation, performed according to three most popular algorithms, has a major or only minor influence on the classification precision in different experimental settings. Hence, three questions are posed in the paper:

- Is the classification precision essentially higher when ICA transformation is performed?
- Has a choice of a specific algorithm for performing ICA an influence on the classification precision?
- Is there any correlation between the classification precision (after performing ICA) and the number of input features used in the classification process?

In order to answer these questions, a set of experiments was planned and performed. In all experiments a data set from the second BCI Competition was used. The results of the experiments, together with a short discussion, are presented in the paper.

2 Independent Component Analysis

The problem of a blind source separation (BSS) consist in finding a matrix W such that the linear transformation will allow to recover the source signals from a set of mixed signals [16-17]. The term 'blind' means that no prior information about the source signals or the mixing process is available [16].

Independent Component Analysis (ICA) is one of the most popular BSS method. ICA problem can be stated as follows. Let's assume that there are n linear mixtures $x_1, \dots x_n$ of n independent components. Vector x (observed signals) can be written as:

$$x = As \tag{1}$$

where A represents a mixing matrix with the size of $n \times n$, and s is the vector of independent components. The aim of ICA is to find a matrix W (i.e. an inverse of the matrix A) to reverse the mixing effect. Then, after computing the matrix W, we can obtain the independent components by [18-19]:

$$y = wX \cong s \tag{2}$$

Most of the popular ICA algorithms put some constraints on the mixed signals. First of them is a statistical independence between source signals s; second, a non-Gaussian distribution of the source signals and the third - the equality of the number of source signals and the number of mixture signals. While two first constrains are main assumptions utilized by many algorithms, the third one is introduced only to decrease the algorithm complexity (it causes that the mixing matrix is square). Furthermore, it is assumed that each source signal has the unit variance $E\{s_i^2\} = 1$. To hold this assumption, the matrix of the source signals is whitened before the ICA calculation [18-19]. One more assumption, introduced only to simplify the algorithm, is that all mixture signals are centered.

As was mentioned earlier, ICA does not require any prior information about the source signals. Instead, ICA algorithms utilize the concept of statistical independency of the mixed signals. According to the formal definition, the variables a and b are said to be independent if information about the value a does not give any information about the value b and vice versa [17], [19]. Technically, independence can be defined in terms of the probability density function (pdf) [18]:

$$f(x_1, x_2, \dots, x_m) = f_1(x_1) f_2(x_2) \dots f_m(x_m) \tag{3}$$

where $x_1, x_2, \dots, x_m$ are random variables.

There are two main approaches to measuring independence: maximization of non-Gaussianity and minimization of mutual information. Most of the existing ICA algorithms are based on one of them. When the first approach is applied, the task for the algorithm is to modify the components in such a way to obtain the source signals of strong non-Gaussian distribution (the assumption is: the stronger non-Gaussianity, the stronger independence [18]). In other words, the distributions of the mixture signals have to be more Gaussian than the source signals. This approach utilizes various

measures of non-Gaussianity, like: kurtosis, negentropy, approximations of negentropy and others [19].

Mutual information, utilized in the second approach, informs how much information about the variable a can be gained from the information about the variable b. Since smaller value of mutual information means that more information about a given system is stored in the variables [18], ICA algorithms based on mutual information approach minimize the mutual information of the system outputs [19].

2.1 FastICA - Deflation Approach

The FastICA algorithm, proposed by Hyvärinen and Oja, is an iterative method to find local maxima of a defined cost function [18-19], [3]. The purpose of this algorithm is to find the matrix of weights w such that the projection $(w^T x)$ maximizes non-Gaussianity [3], [19]. As a measure for non-Gaussianity, simple estimation of negentropy based on the maximum entropy principle is used [18-19]:

$$J(y) \propto [E\{G(y)\} - E\{G(v)\}]^2 \tag{4}$$

where: y – standardized non-Gaussian random variable, v – standardized random variable with Gaussian distribution, $G(.)$ - any non-quadratic function.

There are two classes of FastICA algorithms, the deflation algorithms (called also one-unit algorithms) and the symmetric algorithms [20]. In the deflation approach, the independent components (ICs) are extracted sequentially, one by one. The algorithm can be summarized as follows [19], [21]:

1. Choose an initial vector w (e.g. random)
2. Do steps 3-6
3. $w^+ = E\{xg(w^+x)\} - E\{g'(w^T x)\}w$
4. $w = \frac{w^+}{||w^+||}$
5. Do the Gram-Schmidt orthogonalization:

$$w_{p+1} = w_{p+1} - \sum_{j=1}^{p} w_{p+1}^T w_j w_j$$

$$w_{p+1} = \frac{w_{p+1}}{\sqrt{w_{p+1}^T w_{p+1}}}$$

6. Stop if not converged

Gram-Schmidt procedure, used in the algorithm, prevents different vectors from matrix w from converging to the same maxima [19]. The order, in which the independent components are extracted, depends on the initial value of w.

2.2 FastICA - Symmetric Approach

The only difference between deflation approach and symmetric approach is the procedure of weights calculation. While in deflation approach vectors of weights are calculated one by one, in symmetric approach the estimation of all components (all weights vectors) proceeds in parallel [19-20]. Instead of Gram-Schmidt procedure, the following formula is used in the orthogonalization step:

$$w = (ww^T)^{-1/2}w \tag{5}$$

where w is the matrix of weights vectors $(w_1, \dots, w_n)^T$. The square root of ww^T is obtained from the eigenvalue decomposition of $ww^T = QDQ^T$ as [22]:

$$(ww^T)^{-1/2} = QD^{-1/2}Q^T \tag{6}$$

where Q is the matrix of eigenvectors and D is the diagonal matrix of eigenvalues.

The algorithm is performed until the stop condition (e.g. given by 7 [20]) is met:

$$1 - \min\left(abs\left(diag(w^T w_{old})\right)\right) < \varepsilon \tag{7}$$

where ε is a chosen constant.

2.3 Infomax

Infomax algorithm is based on the general optimization principle for neural networks and other processing systems described by Linsker in 1987 [23]. In general this principle says that a function that maps a set of input values *a* to a set of output values *b* should be chosen or learned so as to maximize the average Shannon mutual information between *a* and *b*. The ICA algorithm utilizing this principle was first proposed in 1995 by Bell and Sejnowski [24] and then in 1997 optimized by Amari [19], [21].

Infomax algorithm for calculating independent components is based on the maximization of the output entropy of a neural network with non-linear outputs [19]. The most essential parameter of this algorithm is a learning rate which does not need to be constant over time and which should give a good compromise between speed of learning and estimation precision [19], [25]. The weights of this neural network are updated according to the following formula [18], [21], [26]:

$$w_{k+1} = w_k + \mu_k[I - 2g(y_k)y_k^T]w_k \tag{8}$$

where: y – matrix of source estimation (y=Wx); k – number of iteration; I – the identity matrix; μ_k – learning rate which may depend on k; $g(.)$ – a nonlinear function.

Mostly a classic logistic function is used as a nonlinear function g [26]:

$$g(y) = \frac{1}{1+e^{-y}}, \tag{9}$$

however, sometimes also its extended version is applied:

$$g(y) = y \pm \tanh(y) \tag{10}$$

Using (9), the Infomax algorithm can be summarized as follows [21]:

1. x = perm(sources);
2. $y = w \times x$
3. $g = \frac{1}{1+e^{-y_k}}$
4. $gu = g \times y^T$
5. $gu = I - 2 \times gu$
6. $w_{k+1} = w_k + \mu_k \times gu \times w_k$

where $perm$ is random permutation.

3 Experimental Settings

The comparison of ICA algorithms described in Section 2 was carried out with a data set submitted to the second BCI Competition (data set III – motor imaginary) by Department of Medical Informatics, Institute for Biomedical Engineering, Graz University of Technology [27]. The data set was recorded from a normal subject (female, 25y) whose task was to control the movements of a feedback bar by means of imagery movements of the left and right hand. Cues informing about the direction in which the feedback bar should be moved were displayed on a screen in the form of the left and right arrows. The order of left and right cues was random. The experiment consisted of 280 trials, each trial lasted 9 seconds. The first 2s was quiet, at t=2s an acoustic stimulus was generated and a cross "+" was displayed for 1s; then at t=3s, an arrow (left or right) was displayed as a cue. The EEG signals were measured over three bipolar EEG channels (C3, Cz and C4), sampled with 128Hz and preliminary filtered between 0.5 and 30Hz. The whole data set, containing data from 280 trials, was then divided into two equal subsets – the first one was intended for classifier training and the second intended for external classifier test. Since only data from the first subset was published with target values (1 - left hand, 2- right hand), only this subset could be used in the research.

In the preprocessing step, the data from the original data set was transformed according to the algorithms described in Section 2. After performing this step, three different sets of signals were obtained:

1. Set of components obtained with FastICA - deflation approach algorithm.
2. Set of components obtained with FastICA - symmetric approach algorithm.
3. Set of components obtained with Infomax algorithm.

All these sets of components, together with the fourth set, composed of original signals from the channels C3, Cz and C4 were used in the experiments.

The data from each set of components, was transformed to a set of frequency band power features. The signal power was calculated separately for:

1. 12 frequency bands: alpha band (8-13Hz) and five sub-bands of alpha band (8-9Hz; 9-10Hz; 10-11Hz; 11-12Hz; 12-13Hz); beta band (13-30Hz) and also five sub-bands of beta band (13-17Hz; 17-20Hz; 20-23Hz; 23-26Hz; 26-30Hz),
2. each of 7 seconds of the trial (data from the first and second seconds of the recordings of each trial were discarded because they covered the period before the clue presentation),
3. each of 3 canals (C3, Cz, C4).

In this way 252 band power features were obtained per each of four sets of components. Taking into account a very small number of trials equal to 140, the number of features had to be significantly reduced before the classification step. In fact with 140 trials, no more than several features should be used without the threat of overfitting. According to Raudys and Jain, at least 10 times more training data per class than the features should be gathered to train the classifier correctly [28].

In order to reduce the number of features, a genetic algorithm, described in details in [8] was used. Some basic features of this algorithm are as follows:

1. An individual is composed of the number of genes equal to the critical number of features, given by the user or calculated automatically in terms of number of observations, number of classes and classifier type.
2. Each gene can take an integer value from the interval $\{0,1...F\}$, where F denotes the dimension of the feature set.
3. The basic genetic operation in the algorithm is a very aggressive mutation. The aggressive mutation means that not only each individual in the population is mutated, but also each gene of each individual.
4. Since, after the mutation a lot of new individuals is born, the selection step is performed after the reproduction step. The selection is made from the population composed of parent individuals and their mutated children.
5. The fitness function is pure classifier accuracy.

As it was stated above, a fitness function of the algorithm used for feature selection was a classification accuracy. A classifier was built per each individual of each generation. Input features introduced to each classifier were encoded in succeeding genes of the evaluated individual.

A linear SVM method was used in the classification process. The classification threshold was set to 0.5 and hence, all classifier results greater than 0.5 were classified as class “2” (right hand) and results smaller or equal to 0.5 were classified as class “1” (left hand). The classifiers accuracy was tested with 10-fold cross-validation. The final accuracy measure of a given feature set was the mean value calculated on the basis of classification accuracy obtained for all validation sets. The accuracy of one validation set was calculated according to the following equation:

$$A_k = \frac{R_k}{U_k} \tag{11}$$

where: A_k - accuracy of k validation subset (k=1...10), R_k - number of properly classified cases from k validation subset, U_k - number of all cases in k validation subset.

4 Results

The experiments were performed with the application prepared for the Matlab 7.12.0 environment. All analyzed preprocessing algorithms, it is: FastICA-deflation, FastICA-symmetric and Infomax, were implemented according to the general schemes given in Section 2. The learning rate in Infomax algorithm was variable in time. In three first iterations it was equal to: 0.01, 0.001 and 0.0001, respectively and in the remaining 500 iterations it was equal to 0.005.

The main aim of the experiments described in the paper was to find out whether the preprocessing with ICA has an essential influence on the classification accuracy. In order to answer this question, the classification accuracy calculated over raw signals and signals preprocessed with the analyzed ICA algorithms had to be compared. The comparison was carried out for different number of input features introduced to the classifier (from one to six). To perform the analysis, six genetic algorithms were prepared. The first one processed individuals composed of only one gene, second processed individuals composed of two genes and so forth up to the last one which processed individuals composed of six genes. Since a genetic algorithm is a heuristic optimization method which gives only sub-optimal solutions, the algorithm was run five times for each from the given six settings. Each from these 120 algorithms (six features, 4 sets of input signals, 5 runs) processed data by 30 generations. After each generation of each genetic algorithm, the individual of the highest value of the fitness function was stored in a table. At the end of the experiments, the results of the best individuals were averaged separately for each set of signals and for each number of input features. The average values of fitness function (classification accuracy) of the best individuals, together with standard deviations are presented in Table 1.

Table 1. The average values of fitness function (classification accuracy) and standard deviation calculated over the best individuals obtained for different sets of signals and different numbers of input features.

	Raw data		Deflation		Symmetric		Infomax	
No. of features	Accuracy [%]	std	Accuracy [%]	std	Accuracy [%]	std	Accuracy [%]	std
1	76.71	0.64	75.14	0.53	75.00	0.45	75.86	0.83
2	82.57	2.7	82.71	1.31	82.43	1.32	82.29	0.29
3	89.00	0.81	88.43	0.53	87.86	1.36	88.57	1.43
4	91.00	1.08	90.86	0.53	90.71	0.9	91.71	0.73
5	92.71	1.06	92.14	1.01	92.57	0.35	92.43	0.57
6	93.14	1.08	92.29	1.53	93.86	0.73	94.57	0.73

A much more compact comparison of results obtained with ICA algorithms and results obtained with raw signals is given in Fig. 1. Each sub-figure of Fig. 1 presents the comparison of the results obtained with raw signals with the results obtained with

signals preprocessed by one of the analyzed ICA algorithms (Fig. 1a - FastICA-deflation, Fig. 1b - FastICA-symmetric, Fig. 1c - Infomax).

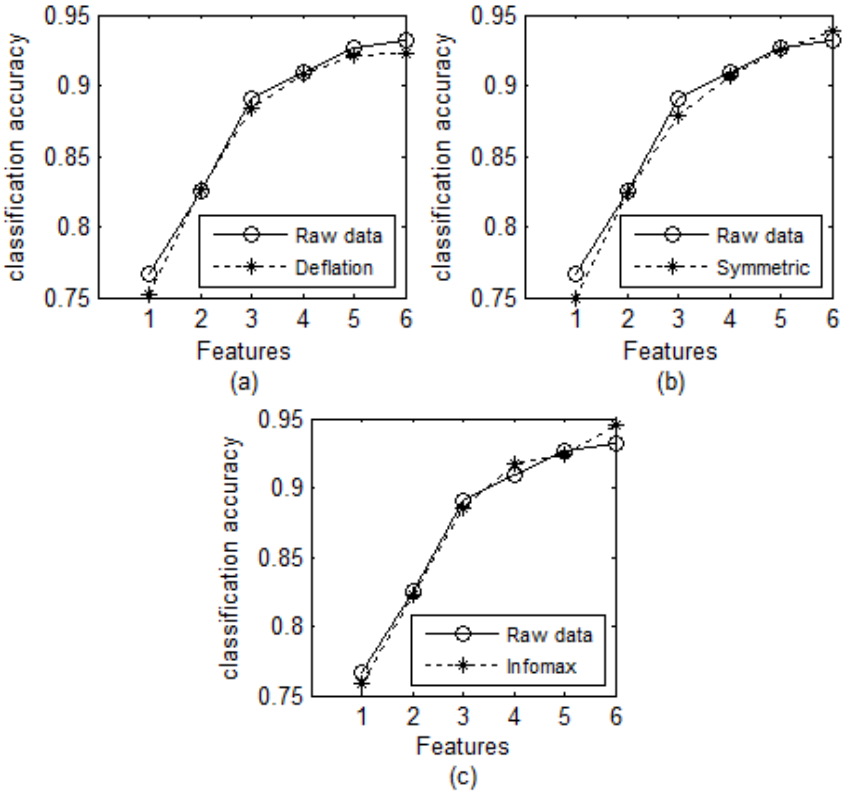

Fig. 1. The average classification accuracy calculated over five runs for each of the six genetic algorithms. Each sub-figure presents the results obtained with raw signals and the results obtained with signals preprocessed by one of the analyzed ICA algorithms (Fig. 1a - FastICA-deflation, Fig. 1b - FastICA-symmetric, Fig. 1c - Infomax).

5 Discussion

In Section 1 three questions regarding the influence of ICA transformation on the classification precision were posed. To answer the first question, about the overall increase in the classification accuracy, the average value of the classification accuracy obtained with all three ICA algorithms was calculated. The result was surprising because the average classification precision calculated for signals preprocessed with ICA algorithms was equal to 87.2% which was slightly lower than the average classification precision calculated for raw signals (87.5%).

In order to answer the second question about the influence of the choice of a specific ICA algorithm on the classification precision, the comparison between algo-

rithms had to be done. This time the results were more consistent with the theory and other analysis. The highest average classification precision for all six genetic algorithms was obtained with Infomax algorithm (87.6%), a slightly smaller precision was obtained with FastICA-symmetric algorithm (87.1) and the smallest precision was obtained with FastICA-deflation (86.9%) (Fig. 2). Looking closer at the average precisions of all three algorithms, one can easily notice that the differences between the performance of ICA algorithms are very small, in fact they are so small that they are not regarded as significant by any reasonable statistical test. Therefore, it is difficult to state that one algorithm is better than the others only on the basis of the classification precision. However, when standard deviations of results are taken into consideration, the conclusion is quite different. The standard deviation of the results gathered in Tab. 1 was on average by 33% lower in case of Infomax (0.8%) than in case of the raw signals (1.2%). This means that the Infomax algorithm gave more stable results and that these results are more reliable than results calculated over raw signals. Also the uncertainty of the classification precision calculated over signals preprocessed with two other ICA algorithms was significantly smaller than in case of raw data (std of FastICA-deflation - 0.9% and std of FastICA-symmetric - 0.9%). Hence, taking into account the standard deviation of results a conclusion of a practical usefulness of ICA transformation should be drawn.

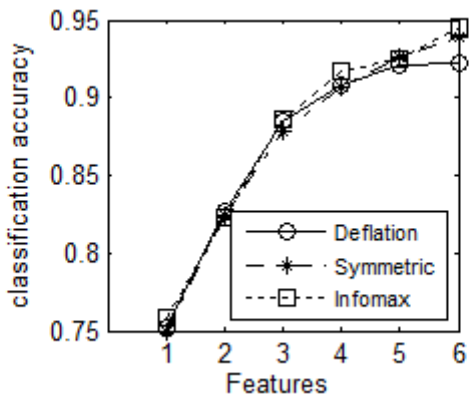

Fig. 2. The comparison of classification accuracy calculated over signals preprocessed with the analyzed ICA algorithms for different numbers of input features.

And finally, the third question posed in Section 1 was about a correlation between difference in the classification precision obtained with signals preprocessed with analysed ICA algorithms and the number of input features used in the classification process. This question is difficult to address because of very small, in fact non-significant differences in classification accuracy obtained over signals preprocessed with all three algorithms at each step of the experiment. On average Infomax gave better results but for different number of input features different algorithms exhibits slightly better performance: FastICA-deflation gave the best result for two features, FastICA-symmetric for five features and Infomax for one, three, four and six features.

6 Conclusion

The classification accuracy is a very important factor of a successful BCI because wrong classification results in executing improper action/command. Misclassification is not allowed, especially when the BCI is used as a system controlling actions of real devices.

The overall goal of this paper was to determine whether the preprocessing with ICA results in increasing the classification accuracy. Since the classification results were very similar with preprocessing and without it, it is very difficult to answer this question decidedly. Undoubtedly, the application of ICA increases the complexity of the whole BCI system so, on the one hand, taking into account only the lack of improvements in the accuracy, the answer should be "no". However, on the other hand, after applying ICA the uncertainty of classification results decreased rapidly which is a very important fact pro ICA application. Hence, before the question posed in the paper will be answer definitely, more experiments have to be done, experiments which will allow to find out which components survive the selection process, whether these components are the same in all ICA algorithms and lastly what is their distribution over the head model.

References

1. Wolpaw J. R., Birbaumer N., McFarland D. J., Pfurtscheller G., Vaughan T. M.: Brain–computer interfaces for communication and control. Clinical Neurophysiology. 113, 767–791 (2002)
2. Wang Y., Shangkai G., Xiaorong G.: Common Spatial Pattern Method for Channel Selection in Motor Imagery Based Brain-Computer Interface. Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference (2005)
3. Oja E., Yuan Z.: The FastICA Algorithm Revisited: Convergence Analysis. IEEE Transaction on Neural Networks, vo. 17. 6, pp. 1370-1381 (2006)
4. Augustyniak P.: Adaptive Wavelet Discrimination of Muscular Noise in the ECG, Proc. Computers in Cardiology (IEEE-EMB), Vol. 33, pp. 481-484 (2006)
5. Yang R., Song A., Xu B.: Feature Extraction of Motor Imagery EEG Based on Wavelet Transform and Higher-Order Statistics, International Journal of Wavelets. Multiresolution and Information Processing, Vol. 8, No. 3, pp. 373-384 (2010)
6. Müller-Putz G. R., Kaiser V., Solis-Escalante T., Pfurtscheller G.: Fast set-up asynchronous brain-switch based on detection of foot motor imagery in 1-channel EEG, Medical & Biological Engineering & Computing, Mar2010, Vol. 48 Issue 3, p. 229-233
7. Koprinska I.: Feature Selection for Brain-Computer Interfaces, T. Theeramunkong et al. (Eds.): PAKDD Workshops 2009, LNAI 5669, pp.100-111, Springer-Verlag Berlin Heidelberg, (2010)
8. Rejer I.: Genetic Algorithms in EEG Feature Selection for the Classification of Movements of the Left and Right Hand, Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013, Advances in Intelligent Systems and Computing, Springer, Vol. 226 (2013)
9. Burduk R.: Recognition Task with Feature Selection and Weighted Majority Voting Based on Interval-Valued Fuzzy Sets in Computational Collective Intelligence. Technologies and

Applications, ed. Nguyen, Ngoc-Thanh and Hoang, Kiem and Jędrzejowicz, Springer Berlin Heidelberg (2012)
10. Pfurtscheller G. Neuper Ch., Schlögl A., Lugger K.: Separability of EEG Singals Recorded During Right and Left Motor Imagery Using Adaptive Autoregressive Parameters, IEEE Trans. on Rehabilitation Engineering, vol. 6, no. 3, September (1998)
11. Bobrowski L., Topczewska M.: Separable Linearization of Learning Sets by Ranked Layer of Radial Binary Classifiers. Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013, Advances in Intelligent Systems and Computing, Springer, Vol. 226, (2013)
12. Wozniak M., Krawczyk B., Combined classifier based on feature space partitioning. Applied Mathematics and Computer Science, No.22 (4): 855-866 (2012)
13. Chen X., Wang L., and Xu Y.: A symmetric orthogonal FastICA algorithm and applications in EEG. ICNC'09, Fifth International Conference on Natural Computation, Vol. 2, (2009)
14. Peterson D. A., Knight J. N., Kirby M. J., Anderson Ch. W., Thaut M. H.: Feature Selection and Blind Source Separation in an EEG-Based Brain-Computer Interface. EURASIP Journal on Applied Signal Processing, 19, pp. 3128-3140 (2005)
15. Boord P., Craig A., Tran Y., Nguyen H.: Discrimination of left and right leg motor imagery for brain–computer interfaces. Medical & Biological Engineering & Computing, Apr2010, Vol. 48 Issue 4, p. 343-350
16. Park H.M., Oh S.H., Lee S.Y.: A modified infomax algorithm for blind signal separation. Science Direct. Neurocomputing 70, 229-240 (2006)
17. Stone J.V.: Independent component analysis: an introduction. TRENDS in Cognitive Sciences Vol.6 No.2, 59-64 (2002)
18. Langlois D., Chartier S., Gosselin D.: An Introduction to Independent Component Analysis: InfoMax and FastICA algorithms, Tutorials in Quantitative Methods for Psychology, 6:31-38 (2010)
19. Hyvärinen A, Oja E.: Independent Component Analysis: Algorithms and Applications. Neural Networks, 13(4-5):411-430 (2000)
20. Tichavský P., Koldovský Z., Oja E.: Performance Analysis of the FastICA Algorithm and Cramér–Rao Bounds for Linear Independent Component Analysis. IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 54, NO. 4, 1189-1203 (2006)
21. Naik G., Kumar D.: An Overview of Independent Component Analysis and Its Applications. Informatykica, 63-81 (2011)
22. Fan C., Wang B., Ju H., A New FastICA Algorithm with Symmetric Orthogonalization, 2006 IEEE, 2058-2061
23. Linsker R., Self-Organization in a Perceptual Network. Computer. Vol. 21 pp. 105 - 117 IEEE (1988)
24. Bell A. J. Sejnowski T. J.: An information-maximisation approach to blind separation and blind deconvolution, Journal Neural Computation. Vol. 7, pp. 1129-1159 (1995)
25. Raimondo F., Kamienkowski J. E., Sigman M., Slezak D. F.: CUDAICA: GPU Optimization of Infomax-ICA EEG Analysis. Computational Intelligence and Neuroscience. Vol. 2012, (2012)
26. Karhunen J.: Neural approaches to independent component analysis and source separation. Proc 4th European Symposium on Artificial Neural Networks (ESANN '96) (1996)
27. Data set III, II BCI Competition, motor imaginary, http://bbci.de/competition/ii/index.html
28. Raudys S. J., Jain A. K.: Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis and Machine Intelligence, 13, 3, 252–264 (1991)