The Effect of Methods Addressing the Class Imbalance Problem on P300 Detection

Guoqiang Xu, Furao Shen and Jinxi Zhao

***Abstract*— This paper studies empirically the effect of different sampling methods on training classifiers on the imbalanced data of the BCI P300 Speller. Both over-sampling and under-sampling are considered. Besides some existing methods like SMOTE that have been shown to be effective in addressing the class imbalance problem we also proposed a new under-sampling technology, namely, instance-remove algorithm which is based on the property of P300 data sets. The classifiers for testing are FLDA and linear SVM. Experimental results suggest that not all of the sampling methods are effective in P300 detection, and even the same method may have different influence on different classifiers. It reveals that the SMOTE technique which is a variant of over-sampling is very effective in training an FLDA classifier while other methods are slightly effective or ineffective both in training FLDA and Linear SVM. The study also suggests that the over-sampling is more effective than under-sampling on both classifiers.**

I. INTRODUCTION

BRAIN-COMPUTER INTERFACE (BCI) aims to enable people especially those who are disabled to communicate with computers and devices directly. It has become one of the most recent fields in Compute Science. Among so many types of BCI signals P300 is one of the most successfully used Event Related Potential (ERP) Electroencephalography (EEG) Signals. The P300 potential occurs approximately 300 ms after the subject is exposed to a certain stimuli, people only need to recognize whether the signals contain P300 potential or not in classification. Researches have used many different methods in P300 detection in order to have a good performance in their system. Some researches used popular linear or nonlinear classifiers like Neural networks [1] [2], Support Vector Machines (SVMs) [3] for classification while others used strong feature extraction techniques like Independent Component Analysis (ICA) [4], Wiener Filter [5], Continuous Wavelet Transform (CWT) [6]. More recent approaches take account of some novel technologies like convolutional neural networks and ensemble learning algorithms in P300 detection [3] [7] [8]. These later proposed algorithms have been proved to have a high character recognition rate with simple data preprocessing procedure respectively.

However, the class imbalance problem of P300 data sets acquired of traditional P300 Speller didn't attract much attention of researchers and it will be solved in a simple way just by random sampling two class samples one by one without replacement from the primal data set when noticed by the researchers. This paper studies the class imbalance problem of P300 data sets acquired of traditional P300 speller paradigm and shows that by reducing the imbalance ratio of P300 data sets one can improve the performance of classifiers like FLDA and linear SVM in P300 detection.

Guoqiang Xu(njuxgq@163.com), Furao Shen(frshen@nju.edu.cn) and Jinxi Zhao(jxzhao@nju.edu.cn) are with the National Key Laboratory for Novel Software Technology, and Department of Computer Science and Technology at Nanjing University, Nanjing, 210046, P.R.China.

According to Gu et al. [9], there are two main approaches considering data-level and algorithm level in dealing with imbalanced data sets. The data-level methods re-balance the ratio between different classes by changing the training data distribution and then train classifiers on the new balanced or lower imbalanced ratio data sets, while the algorithm level methods strengthen the existing classifiers by adjusting algorithms to pay more attention to the smaller classes. This paper studies methods that have been shown to be effective in addressing the class imbalance problem from data-level, applied to P300 data. Under-sampling and over-sampling methods are two main methods to re-balance the data sets from data-level. The under-sampling methods sample smaller classes and bigger classes one by one from the primal training samples so that the ratio of each class is balanced. It is obvious that the new training data is a subset of the original training data and the samples that are not selected to the new training data are useless and meanwhile some important information may be lost, which is a main problem of under-sampling methods. Contrast to the under-sampling methods, the over-sampling methods increase the the number of samples of smaller classes by some special algorithms like SMOTE or simply replicate smaller samples etc.

In particular, this paper studies empirically the effect of over-sampling and under-sampling in training classifiers on the imbalanced data of the BCI P300 Speller. Besides some existing methods that have been shown to be effective in addressing the class imbalance problem we also proposed a new under-sampling technology, namely, sample-remove which is based on the property of P300 data. Note that all the algorithms studied in this paper are about two-class classification problem and the smaller class is positive class. The classifiers Fisher Linear Discrimination Analysis and Linear Support Vector Machine are tested for classification because they are two main kinds of linear classifiers which are probably the most popular algorithms for BCI applications [10]. Experimental results suggest that not all of the sampling methods are effective in P300 detection, and even the same method may have different influence on different classifiers. It reveals that the SMOTE technique which is a variant of over-sampling is very effective in training an FLDA classifier on the P300 data while other methods are slightly effective or ineffective both in training FLDA and

Linear SVM. The study also suggests that methods that are proved effective in the experiment always perform better in training FLDA than in training Linear SVM on the P300 data sets. What's more, the over-sampling is more effective than under-sampling on both classifiers.

This paper has been organized as follows. Section II describes the P300 Speller Diagram and data sets used in our experiments, section III explains learning methods, section IV shows the set of our experiments and results obtained and section V is the conclusion and discussion.

II. Data acquisition

A. P300 Speller Diagram and Data set

The data used in this article is the 3nd Wadsworth BCI Data set from BCI competition 2005 [11] which is based on the P300 speller paradigm. P300 Speller Diagram is first designed by Farwell and Donchin [12] in 1988 which is based on the so-called oddball paradigm which states that rare expected stimuli produce a positive deflection in the EEG after about 300 ms. In a P300 Speller Diagram the subject is presented with a 6 by 6 matrix of 36 characters as illustrated in Figure 1. For the spelling of a single character, each of the 12 rows and columns of the matrix is then intensified according to a random sequence. The subject is asked to focus his attention on the character he wants to spell and then a P300 evoked potential appears in the EEG in response to the intensification of a row or column containing the desired character. Usually, P300 potential is hard to detect with only one repetition so one need to repeat a character trial several times to make the spelling procedure more reliable. Hence, high recognition rate with few sequence repetitions is one of the most important targets for a BCI system.

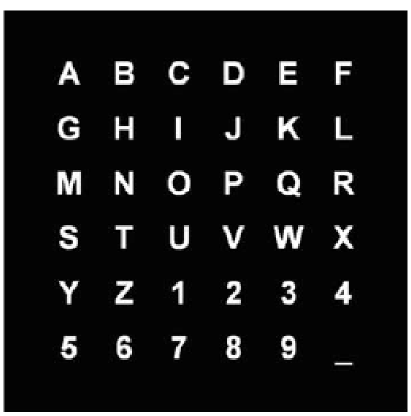

Fig. 1. Stimulus matrix from the P300 Speller Diagram [13].

In BCI Competition III, data has been recorded by 64 channels from two different subjects and five different sessions. Each session is composed of runs, and for each run, a subject is asked to spell a word. For a given acquisition session, all EEG signals of 64 channel scalp have been continuously collected. The sequence of intensification is repeated 15 times for each character to spell. Before digitization at a sample rate of 240 Hz, Signals have been band-pass filtered from 0.1-60Hz [13].

B. Preprocessing and Feature Extraction

Since not all the EEG signals acquired after each intensification are useful and interesting, we divide the continuous signal into segments for each channel and choose the segments that are closely related to decide whether the signal contain P300 potential or not. We extract all signals between 0 and 667 ms posterior of each intensification from all channels of each character trial to make sure that this window is large enough to capture all required time features for an efficient classification according to the knowledge that the P300 potential appears about 300ms after the stimulus. Afterwards, in order to eliminate high frequency and low frequency noise, each extracted signal has been filtered with an 8 order band-pass Chebyshev Type I Filter with 0.1 Hz low cutoff and 20 Hz high cutoff frequency and then the sample rate has been decimated to 20Hz. Finally, the dimension of each sample is 896($64 \times 20 \times 0.667$).

For a single subject, the training set is made up of 85 characters spelling corresponding to 15300=$12\times15 \times 85$ post-stimulus labeled signals and the the testing set is made up of 100 characters spelling corresponding to 18000=$12 \times 15 \times 100$ post-stimulus unlabeled signals. In the training set of each subject, 2550 of the instances containing P300 potential while left 12750 instances don't as two out of twelve intensification contain P300 potential in a trail. Label the instances contain P300 potential positive and others negative. Hence, the ratio of negative class and positive class is is 5:1 which means the P300 training data is a naturally imbalanced data set.

III. Learning Methods

A. Over-Sampling

Over-sampling has been shown is effective in learning with imbalanced data sets [9] [14], which changes the training data distribution by resampling the small class with some special methods. Note that over-sampling usually increase the training time and may lead to overfitting sice it involves making exact copies of examples [15] [16]. This paper studies two over-sampling methods, namely, random over-sampling and SMOTE [16].

1) Random Over-Sampling: Random over-sampling is a popular method in solving the class imbalance problem, which resamples the small class by random sampling with replacement until it contains as many examples as the other class. The random over-sampling considering two-class classification problem is shown in $Algorithm$ 1.

2) SMOTE: SMOTE [16] algorithm is a heuristic over-sampling method which resamples the small class rather than simply replicate small class through taking each small class sample and introducing synthetic examples along the line segments joining its small class nearest neighbors. The algorithm can avoid the over-fitting problem and has been proved to be a successful over-sampling technique in literature [17]. The SMOTE algorithm is described in the $Algorithm$ 2.

Algorithm 1 Random Over-sampling Algorithm

Input: Original training set D

The positive (smaller class) set P in D

Output: New training data D^*

1: Put all the original training instances into D^*.
2: Compute: $N^* = N_{neg} - N_{pos}$, where N_{neg} is the number of negative instances in D and N_{pos} is the number of instances in P.
3: Resample N^* number of positive instances from P and put them into D^*

Algorithm 2 SMOTE Algorithm

Input: Original training set D

The positive (smaller class) data set P in D

Output: New training set D^*

1: Put all the instances of D into D^*
2: $k =$ Number of nearest neighbors.
3: For each sample x in P.
4: Find the k-nearest neighbors (smaller class instances) of x in P and then choose one of them randomly. Let y be the chosen example.
5: Compute: $diff = x - y$.
Compute: gap = random number between 0 and 1.
Compute: $x^* = diff * gap + x$, and then x^* is the new generated example.
6: Add x^* to D^*.
7: End for

B. Under-Sampling

Under-sampling is another method which is also effective in learning with imbalanced data. It samples smaller classes and bigger classes one by one from the primal training set so that the ratio of each class is 1:1. It is obvious that the new training set is a subset of the original training set and the instances that are not selected to the new training set are useless and meanwhile some important information may be lost. This paper studies two under-sampling methods, namely, random under-sampling and instance-remove.

1) random under-sampling: Random under-sampling is a very simple method to solve the class imbalance problem and the random under-sampling considering a two-class classification problem is shown in $Algorithm$ 3.

Algorithm 3 Random Under-sampling Algorithm

Input: Original training set D

The positive (smaller class) set P in D

The negative (bigger class) set N in D

Output: New training data D^*

1: Put all the instances of P into D^*.
2: Random sampling N_{pos} number of negative instances without replacement from P and put them into D^*, where N_{pos} is the number of instances in P.

2) Instance-Remove: This method is an under-sampling technology by removing a large amount of negative instances and few positive instances from the P300 data set, meanwhile, reduce the redundancy of the data set. Have careful studied the signal preprocessing procedure, we find the extracted signal segments of two adjacent intensifications are partly overlapping. For spelling a character in the run, each row and column in the matrix was randomly intensified for 100ms. After intensification of a row/column, the matrix was blank for 75ms [13]. Since we extract all data instances between 0 and 667 ms posterior of each intensification, the time overlapped of two adjacent signal segments is 492ms that is 73.76% of total 667ms. This means features of a instance may be similar to others that extracted from its adjacent intensifications in the same block. Note that the above property may be unavoidable because only rare expected stimuli can produce a clear P300 potential in the EEG after about 300 ms according to the odd ball paradigm. Nowadays many researchers extracted time domain features of P300 data by dividing signals into segments: Li Yandong [11] extracted signal segments between 100 and 850 ms posterior of each intensification, Rakotomamonjy extracted signal segments between 0 and 667 ms posterior of each intensification [3]. Hence, It is interesting and necessary to study the overlap of P300 signal segments. As signal of an intensification may contain P300 potential or not, we can divide the overlap of two adjacent signal segments into three cases based on the labels of the extracted instances:

1. Both instances are negative.
2. One instance is negative and another one is positive.
3. Both instances are positive.

Instances in case two and cases three may be noises because in these two case signal overlapping will have much influence on the instances contain P300 so that these instances could be removed from the training data. More over, by removing the instances that are noises, the ratio of number between negative class and positive class is reduced. In order to remove those noises we proposed instance-remove algorithm which is described in the $Algorithm$ 4.

Algorithm 4 Instance-remove Algorithm

Input: Training set D.

Output: New training set set D^*

1: Compute threshold $C_{threshold} = arccos\left(\frac{\langle m_p, m_n\rangle}{\|m_p\|\cdot\|m_n\|}\right)$, where m_p is the mean vector of positive instances in D and m_n is the mean vector of negative instances in D.
2: For each positive instance x in D.
3: Let instance y be the adjacent instance of x in the same trial.
4: Compute: $C = arccos\left(\frac{\langle x, y\rangle}{\|x\|\cdot\|y\|}\right)$
5: If $C < C_{threshold}$, then remove instance y from D.
6: End for
7: Put all the instances of D into D^*

C. Character Recognition

When a classifier f has been trained, one can get classifier output of each testing instance by the decision function $f(x)$.

Afterwards, averaging classifier's outputs over the epochs are presented to recognize characters. Note that this idea was first applied by Kaper et al. [19]. We combine epochs by summing the decision function values from corresponding rows/columns from different epochs. Then row/columns with the maximal total value after n epochs is considered to the desire row/columns that contain P300 potential and then the character is recognized according to the P300 Speller matrix.The coordinate of the character are defined as follow:

$$S_{r|c} = \arg\max_{i} \left(\frac{1}{n} \sum_{k=1}^{n} f(x_{r|c}^{k}) \right)$$

where i=1,...6, and $x_{r|c}^{k}$ is the sample of the given row/column during kth epoch, n is the number of epoches. Hence, as the epochs increase the decision value of each trial is averaged to determine the desire row and column.

IV. Experiment and Result

Before training classifiers, the optimal parameter C of linear SVM should be selected. The cross validation method is used to select the optimal parameter C of linear SVM, finally we set the parameter C=0.001 for both subjects. Note that the optimal parameters of different subjects may be different. Because of the instability of SMOTE, both random over-sampling and under-sampling, ten repeated experiments will be done and the average result of them is used as the final result.

The classification result of FLDA and SVM after applying SMOTE algorithm and random over-sampling technology is shown in Table I and II. Note that the ratio of negative class and positive class is reduced to 1:4 which means 2550 positive instances are added to the training data for each Subject. The ratio is not reduced lower than 1:4 because in the experiment we find there is no obviously improvement if we add more positive instances into the new training data. From the result we see that after applying SMOTE algorithm the performance of FLDA is improved obviously while almost no change of linear SVM. The result also shows that random under-sampling is ineffective both on FLDA and Linear SVM.

TABLE I

Character Recognition Rate for Linear SVM before and after applying over-sampling and under-sampling algorithms.

Subject	Method	Epoch 1	2	3	4	5	10	13	15
A	SVM	16	35	54	64	66	89	95	98
	SMOTE	18	38.9	58.6	62.9	67.5	87.6	96.5	97.7
	over-sampling	18	34	54.7	60.3	64.4	86.8	95.9	97.8
	instance-remove	15	32	55	58	64	88	96	98
	under-sampling	16.1	32.6	51	59.5	62.7	85.6	93.1	95.9
B	SVM	42	63	69	72	80	94	96	96
	SMOTE	42	61.7	68.8	70.9	79.5	94	95.9	96.3
	over-sampling	42.3	62.2	68.3	73.3	79.1	92.2	93.5	95.1
	instance-remove	41	62	67	74	80	94	96	97
	under-sampling	37.8	56.4	68.3	71.6	78.2	93.9	96.7	96.5
Mean	SVM	29	49	61.5	68	73	91.5	95.5	97
	SMOTE	30	50.3	63.7	66.9	73.5	90.8	96.2	97
	over-sampling	30.15	48.1	61.5	66.8	71.75	89.5	94.7	96.45
	instance-remove	28	47	61	66	72	91	96	97.5
	under-sampling	26.95	44.5	59.65	65.55	70.45	89.75	94.9	96.2

TABLE II

Character Recognition Rate for FLDA before and after applying over-sampling and under-sampling algorithms.

Subject	Method	Epoch 1	2	3	4	5	10	13	15
A	FLDA	22	33	45	53	58	85	93	93
	SMOTE	17	35.4	51.7	58.9	62.6	88.9	95.5	96.6
	over-sampling	19.9	32.2	44.4	51	56.3	85.9	92.5	92.6
	instance-remove	18	32	45	50	59	85	92	92
	under-sampling	18.1	26	37.7	45.5	51	81	89	90.9
B	FLDA	37	62	67	69	75	88	89	92
	SMOTE	41.7	62.2	68.1	71	78.4	92.4	93.5	94.8
	over-sampling	37.4	60.8	65.6	69.2	76.2	87.3	90.3	92.6
	instance-remove	35	56	64	68	73	86	91	91
	under-sampling	36.6	56.5	63.6	67.1	72.2	86.5	89.4	89.9
Mean	FLDA	29.5	47.5	56	61	66.5	86.5	91	92.5
	SMOTE	29.35	48.8	59.9	64.95	70.5	90.65	94.5	95.7
	over-sampling	28.65	46.5	55	60.1	66.25	86.6	91.4	92.6
	instance-remove	26.5	44	54.5	59	66	85.5	91.5	91.5
	under-sampling	27.35	41.25	50.65	56.3	61.6	83.75	89.2	90.4

TABLE III

The training data size after applying instances remove algorithm for each subject.

	SubjectA	SubjectB
P300	2138	2136
no P300	8659	8689

The training data size after applying instance-remove algorithm for each subject is shown in Table III.

We see the number of instances that are removed from training data is about 26%-28% of the number of original training data for both subjects. Meanwhile, the unbalance ratios of both data set are also reduced. Though so many instances are removed the recognition rate of characters of both linear SVM and FLDA on both testing data is slightly changed while the training time is reduced greatly. Hence, experimental results demonstrate that the instances remove algorithm can remove useless instances efficiently and at the same time reduce training time.

V. Conclusion

In this paper, the data sets of BCI P300 Speller are careful analysed and we find that the P300 data sets are nature imbalance data. Then the effect of sampling including over-sampling and under-sampling are studied empirically on the the P300 data sets. Experimental results suggest that not all of the sampling methods are effective in P300 detection, and even the same method may have different influence on different classifiers trained on P300 data. It reveals that the SMOTE technique which is a variant of over-sampling is very effective in training an FLDA classifier and the instances remove algorithm could reduce the training time without changing the character recognition rate of FLDA and SVM while other methods are slightly effective or ineffective both in training FLDA and Linear SVM in P300 detection. The study also suggests that methods that are proved effective in the experiment always perform better in training FLDA than in training Linear SVM on the P300 data sets. A probable reason is that FLDA is more sensitive to the change of data distribution than linear SVM. What's more, the over-sampling is more effective than under-sampling on both classifiers.

However, the data we used in the paper only contains data sets of two subjects, hence further research should be down to investigate the behavior of more subjects. Moreover, we think more works need to be down to make deep study on the property of the data set of P300 Speller.

ACKNOWLEDGEMENTS

This work was supported in part by the 973 Program 2010CB327903, the Fund of the National Natural Science Foundation of Jiangsu NSF grant BK2011567.

The authors would like to thank reviewers for their helpful comments on the paper.

REFERENCES

[1] E. Haselsteiner and G. Pfurtscheller, "Using Time Dependent Neural Networks for EEG Classification", *IEEE Trans. Rehabilitation Eng.*, vol. 8, pp.457–463, 2000.

[2] N. Masic and G. Pfurtscheller, "Neural Network Based Classification of Single-Trail EEG Data", *Artificial Intelligence in Medicine*, vol. 5, pp. 503–513, 1993.

[3] A. Rakotomamonjy and V. Guigue, "BCI Competition III: Data Set II Ensemble of SVMs for BCI p300 Speller", *IEEE Trans. Biomedical Eng.*, vol. 55, pp. 1147–1154, Mar. 2008.

[4] Xiaorong Gao and Bo Hong and Xiaobo Miao and Shangkai Gao and Fusheng Yang, "BCI competition 2003-data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications", *IEEE Trans. Biomedical Eng*, vol. 51, pp. 1067–1072, 2004.

[5] Min Ki Kim and Sung-Phil Kim, "Detection of P300 Components Using the Wiener Filter for BCI-based spellers", *Proceedings of 2011 8th Asian Control Conference*, pp. 892–896, May. 2011.

[6] V. Bostanov, "BCI competition 2003-data sets Ib and IIb: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram", *IEEE Trans. Biomedical Eng*, vol. 51, pp. 1057–1061, 2004.

[7] Hubert Cecotti and Axel Graser, "Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces", *IEEE Transaction On Pattern Anylysis and Machine Intelligence*, vol. 33, pp. 433–445, 2011.

[8] Ulrich Hoffmann and Gary Garcia and Jean-Marc Vesin and Karin Diserens and Touradj Ebrahimi, "A Boosting Approach to P300 Detection with Application to Brain-Computer Interfaces", *Neural Engineering*, 2004.

[9] Qiong Gu and Zhihua Cai and Li Zhu and Bo Huang, "Data mining on imbalanced data sets", *International Conference on Advanced Computer Theory and Engineering*, pp. 1020–1024, Dec. 2008.

[10] F lotte, M congedo, A Lecuyer, F Lamarche and B Arnaldi, "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces", *Journal of Neural Engineering*, No. 4, 2007.

[11] B. Blankertz, BCI Competition III Webpage. [Online]. Available:http //ida. first.fraunhofer.de/projects/bci/competition_iii. 2005.

[12] L. Farwell and E.Donchin, "Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials", *Electroencephal. Clin. Neurophysiol.*, vol. 51, pp. 1034-1043, 2004.

[13] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, "BCI2000: A general-purpose brain-computer interface(BCI) system", *IEEE Trans. Biomed. Eng.*, vol. 51, pp. 1034-1043, 2004.

[14] N.Japkowicz and S.Stephen,"The class imbalance problem: a systematic study", *Intelligent Data Sets*, Austin, pp. 10–15, 2000.

[15] C.Drummond and R.C.Holte, "C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling", *Working Notes of the ICML'03 Workshop on Learning from Imbalance Data Sets*, Dce, 2003.

[16] N. V. Chawla and K W Bowyer and L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique", *Journal of Artificial Intelligence Research*, vol. 16 pp. 321–357, Dec. 2002.

[17] G. E. A. P. A. Batista and R. C. Prati and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data", *SIGKDD Explorations Newsletter*, vol. 6 pp. 20–29, June. 2004.

[18] http://bbci.de/competition/iii/results/index.html, *Results of BCI Competition III*, 2005.

[19] Matthias Kaper, Peter Meinicke, Ulf Grossekathoefer, Thomas Lingner and Helge Ritter, "BCI Competition 2003−Data Set IIb: Support Vector Machines for the P300 Speller Paradigm", *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, Vol. 51, No. 6, June 2004.