RESEARCH ARTICLE

Fixed distance neighbour classifiers in brain computer interface systems

Asiri Nanayakkara* and Zahmeeth Sakkaff
Artificial Intelligence and Applied Electronics Research Unit, Institute of Fundamental Studies, Hanthana Road, Kandy.

Abstract: A new classification method, which is closely related to k nearest neighbour (kNN) classification method is introduced for identifying cognitive tasks in brain computer interface (BCI) systems. This new method is named fixed distance neighbour (FDN) classifier. Performance of the FDN method is tested with feature vectors derived from EEG datasets recorded for imagery motor movement mental tasks. For comparison purposes, performance of kNN classification method is also tested with the same feature vectors. It was found that FDN performed slightly better than kNN for most of the datasets used in this study, indicating that FDN is a viable classification method, which can be used in place of kNN in BCI systems.

INTRODUCTION

Brain computer interface (BCI) technology is an emerging technology, which enables people with motor disabilities to communicate with the external world. It does not require any peripheral muscular activity. The user of this technology can send commands to communicate or control electronic devices by means of brain activity alone.

In order to control a BCI system, the user must produce different brain activity patterns, which could be identified by the system and translated into commands. In most current BCI systems this identification is carried out by a classification algorithm, which automatically recognizes the class from data as represented by a feature vector. Usually, in EEG based BCI systems, the EEG signal is first preconditioned to remove undesired frequencies. Even after preconditioning, the signal cannot be directly used as a feature vector for classification due to its high dimensionality. Therefore, low dimensional feature vectors are constructed using various feature vector construction techniques. Feature vector construction methods used in BCI systems include amplitude values of EEG signals (Kaper *et al.*, 2004); band powers (BP) (Pfurtscheller *et al.*, 1997); power spectral density (PSD) values (Millán & Mourino, 2003; Chiappa & Bengio, 2004); autoregressive (AR) and adaptive autoregressive (AAR) parameters (Pfurtscheller *et al.*, 1998; Penny *et al.*, 2000); time frequency features (Wang *et al.*, 2004); inverse model-based features (Qin *et al.*, 2004; Kamousi *et al.*, 2005; Congedo *et al.*, 2006); Bi-scale wavelets (Mason & Birch, 2000) and PSD with Welch's periodogram (Barreto *et al.*, 2004).

After constructing feature vectors, suitable classification techniques are used to classify EEG signals. For EEG based BCI systems, commonly used classification algorithms are linear discriminant analysis (LDA), support vector machines (SVM), neural networks (NN), nonlinear Bayesian classifiers and nearest neighbour classifiers (Lotte *et al.*, 2007). There are several versions of nearest neighbour classifiers for solving general classification problems (Domeniconi *et al.*, 2002; Voulgaris & George, 2008).

The efficiencies of these different classifiers are found to be strongly dependent on the kind of BCI used. As an example, Gaussian SVM, LDA, hidden Markov model (HMM), finite impulse response neural network (FIR NN) and Bayes quadratic integrated over time have shown good performances (83 % - 89 %) for pure motor imagery based BCI (Lotte *et al.*, 2007). On the other hand, in movement intention based BCI, k nearest neighbours (kNN), multilayer perceptron (MLP), SVM

---

* Corresponding author (asiri@ifs.ac.lk)

and LDA have shown better performance than other classifiers (Lotte *et al.*, 2007). However, kNN algorithms have not shown very good performance for asynchronous BCI (Lotte *et al.*, 2007).

For most of the BCI systems, the system has to be trained and parameters of feature vector construction and classification schemes have to be determined at the training stage.

In this paper, a fixed distance neighbour (FDN) classifier, which is closely related to kNN is proposed for BCI systems. Its performance with motor imagery based BCI data and the performance of kNN classifier on the same dataset is also presented.

METHODS AND MATERIALS

Fixed distance neighbour classifier

In kNN classifier, a decision is made by examining the labels on the k nearest neighbours and taking the majority vote. In order to find the k nearest neighbours for a given test feature point, distances between the test feature point and all the training feature points are calculated and sorted. The metric used for determining distances will depend on the problem of interest. The most commonly used metrics are Euclidian distance metric, cityblock distance metric and correlation distance metric. In the FDN classifier introduced in this paper, a hypersphere of fixed radius (say r_c) centered at the test point is considered and labels of the training points inside the hypersphere are examined. The decision is taken according to the majority vote.

Let x_1, x_2, ... x_M be the training set of N dimensional feature vectors and a be a testing feature vector of the same dimension.

$$x_k = \begin{bmatrix} x_{k,1} \\ x_{k,2} \\ . \\ . \\ x_{k,N} \end{bmatrix} \text{ and } a = \begin{bmatrix} a_1 \\ a_2 \\ . \\ . \\ a_N \end{bmatrix}$$

Suppose that the distance between test vector a and i^{th} training vector x_i is $d_{a,i}$, $i = 1, 2, \ldots M$. Note that $d_{a,i}$ can be calculated with any metric as mentioned above. In FDN feature vectors, which participate in classifying the test vector a belongs to the set S_a.

$$S_a = \left\{ x_k | d_{a,k} < r_c \quad 1 \le k \le M \right\}$$

where, $d_{a,k} = (x_k - a)^T (x_k - a)$ for Euclidean metric. The class label of the test vector is predicted as the label of the class, which is having the highest number of feature vectors in S_a.

The major difference between kNN and FDN methods can be described as follows. In kNN, since the number of nearest neighbours are fixed, the radius of the hypersphere, which encloses the nearest neighbours can be very different from one test point to another as shown in Figures 1*(a)* and 1*(b)*. On the other hand, in FDN, since radius r_c of the hypersphere is fixed, the number of neighbours participating in voting can be different from one test point to another as shown in Figures 2*(a)* and 2*(b)*.
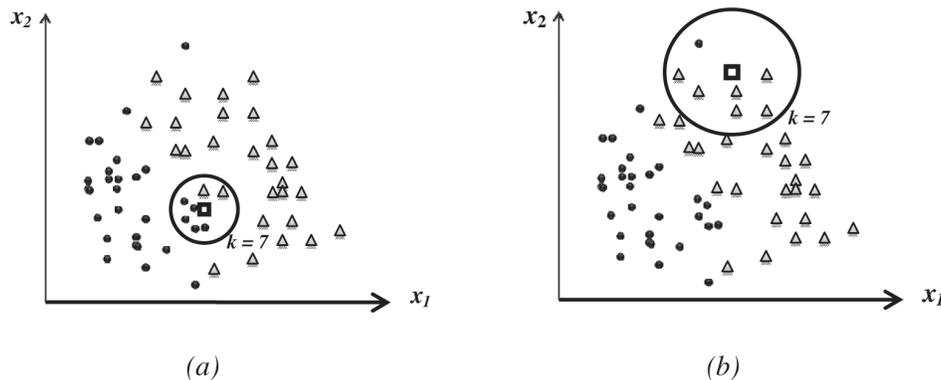

(a)   (b)

Figure 1:   Training feature vector points of a two class problem are represented as Δ and ●. Test point is represented as ▣. (a) In kNN classification, when the concentration of points surrounding the test point is large, a sphere with small radius could enclose seven nearest neighbours. (b) When the concentration of the feature vectors is low, a sphere with larger radius is needed to enclose the seven nearest neighbour points.
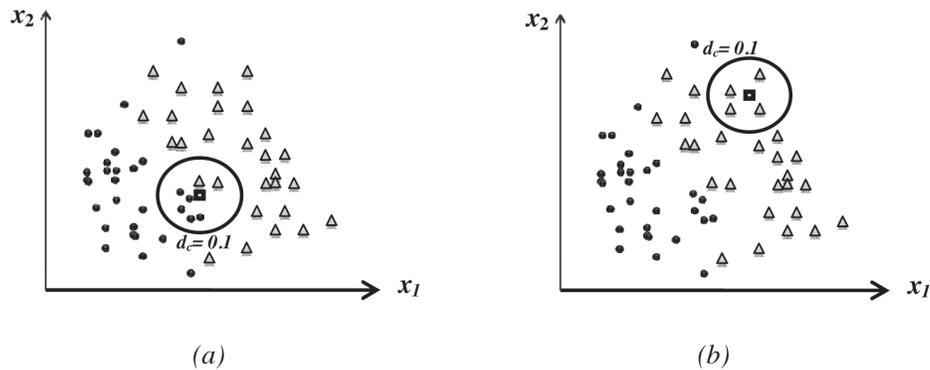
Figure 2: Training feature vector points are as in Figure 1. (a) In FDN classification, when the concentration of points surrounding the test point is large, the sphere with radius d_c encloses large number of neighbours. (b) When the concentration of the feature vectors is low, the sphere encloses fewer numbers of neighbour points.

The radius of the hypersphere, which encloses k nearest neighbours depends on the density of the training feature vectors in the neighbourhood of the test point. Consequently, this radius can vary considerably from one test point to another as mentioned earlier. When k is even, there is a possibility of getting a tie and in such a situation; tie-break is achieved by inclusion of a random point or the next nearest neighbour point for voting. With a sufficiently high value of k and enough training samples, kNN can approximate any function, which enables it to produce nonlinear decision boundaries.

In FDN, for an arbitrary radius centered at the test point, there is a possibility that hypersphere encloses no test feature vectors or encloses all the test feature vectors. Therefore, the radius r_c has to be determined at the training stage such that the radius is large enough to enclose at least the desired number of feature vectors of the training dataset and classification efficiency is maximal. In addition, there is also a possibility that the number of test feature vectors inside the hypersphere is even and can produce a tie for a given test vector. The tie break in FDN can also be carried out by a mechanism as described before for the kNN classifiers. Since optimal k in the kNN method and r_c in the FDN depend on the problem of interest, these parameters are determined empirically with the training dataset.

There are two standard techniques, which can be used for determining the parameters of the classification method. In both techniques, while varying the parameters of the classification method, the performance of the system is evaluated and the optimal values of the parameters are obtained. As described below, these two techniques differ in the way the performance is evaluated.

In the first technique, evaluation of the performance of the classification method is carried out by choosing a single feature vector as a test vector from the training dataset and training the system with the remaining training feature vectors. Then the classification method is tested with this single test vector. This is continued by choosing a different feature vector from the training dataset at a time, until every vector in the training data set has participated. At the end, the total performance is calculated. The optimal parameters are determined by varying the parameter values and repeating the calculation. The parameters, which produce the best performance are taken as the optimal parameters.

In the second technique, part of the training dataset is separated as testing data for determining the optimal parameters, while the remaining part of the training dataset is used for training the system. Performance of the classification method was evaluated for the testing dataset while changing the values of the parameters. The parameters, which produced the best performance are taken as the optimal parameters. These optimal parameters are then utilized for evaluating the overall performance of the classification method.

Performance of the FDN

Performance of the FDN was evaluated by using the EEG dataset recorded while subjects were performing imagery motor movements (IMM) mental tasks (Zachary *et al.*, 1990; Pfurtscheller *et al.*, 1998; Birbaumer, 1999; Wang *et al.*, 2004; Pfurtscheller *et al.*, 2006; Lehtonen *et al.*, 2008) and baseline (BL). The baseline signal represents the mental stage of the subject when he/she is not thinking of any specific mental task without eye

blinks. The IMM mental tasks are the most popular mental tasks used in BCI due to their good performance. First the EEG data was filtered using bandpass filtering and feature vectors were constructed with band power. For comparison purposes, performance of kNN was also determined for the same feature vectors and shown in the Tables.

IMM consists of two mental tasks, namely, imagery of Left middle finger movement (LFM) and imagery of Right middle finger movement (RFM). EEG recordings for IMM were carried out as follows.

The subjects were seated in an armchairs in front of a white blank screen, which was placed approximately one and a half meters away from them. The main reason for placing this white blank screen in front of the subject was to reduce distractions. They were asked not to pay any attention to the white board but to concentrate on mental task they are performing. During the recording of trials, they were instructed to avoid eye movements and to keep their arms and hands relaxed. If eye blinks or eye moments occurred during the recordings, recorded data in the trial was discarded and the trial was repeated to avoid artifacts in EEG data.

RESULTS

In this investigation 24 feature vector datasets generated from two subjects were used. The parameters used in recordings of the two EEG datasets are as follows (Table 1).

In order to compare the performance of FDN and kNN classifiers, twenty four different feature vector datasets were constructed from two subjects by changing the channels, bandpass filter frequencies and width of frequency bands as shown in Tables 2 and 3.

The procedure for selecting parameters to get the best performance for FDN is as follows.

(a) The sets of all the trials were grouped into two; one group for training and the other for testing.
(b) In order to select optimal parameters, only a part of the training group was used for training the system.
(c) With FDN as a classifier, the other part was used for testing the performance while changing the values of the parameters (channels, bandpass filter frequencies and width of frequency bands). The parameters corresponding to the best performance were selected as the optimal parameters.

Table 1: Parameters and settings used in all the EEG recording sessions

Channels used for recording	All 20 channels according to 10 −20 system
Sample rate	256 samples per second
Block size	768 bytes per block per channel
Recording duration	9 seconds

Table 2: Parameters for datasets D1 - D12

Dataset label	Channels	Filter frequencies (low, high)	Width of frequency bands
		BL *vs*. LFM	
D1	1, 8, 11	(1, 30)	40
D2	5, 10, 15	(1, 34)	40
D3	10, 15, 19	(1, 39)	41
D4	3, 7, 10	(1, 32)	42
D5	3, 7, 15	(1, 31)	41
D6	7, 10, 15	(1, 31)	40
		BL *vs*. RFM	
D7	1, 8, 11	(1, 33)	36
D8	5, 10, 15	(1, 31)	39
D9	10, 15, 19	(1, 30)	36
D10	3, 7, 10	(1, 30)	40
D11	3, 7, 15	(1, 40)	38
D12	7, 10, 15	(1, 32)	40

Table 3: Parameters for datasets D13 – D24

Dataset label	Channels	Filter frequencies (low, high)	Width of frequency bands
		BL *vs*. LFM	
D13	1, 8, 11	(1, 31)	41
D14	5, 10, 15	(1, 38)	40
D15	10, 15, 19	(1, 32)	40
D16	3, 7, 10	(1, 33)	32
D17	3, 7, 15	(1, 35)	38
D18	7, 10, 15	(1, 35)	38
		BL *vs*. RFM	
D19	1, 8, 11	(1, 32)	40
D20	5, 10, 15	(1, 39)	39
D21	10, 15, 19	(1, 33)	39
D22	3, 7, 10	(1, 36)	40
D23	3, 7, 15	(1, 41)	38
D24	7, 10, 15	(1, 44)	31

Table 4: Performance of FDN and kNN for datasets D1 – D12

Dataset	FDN		kNN	
	d_c	Performance	k	Performance
1	0.1690	93 %	3	92 %
2	0.1990	97 %	9	95 %
3	0.1770	93 %	7	93 %
4	0.2270	95 %	15	95 %
5	0.1760	95 %	3	93 %
6	0.1760	90 %	7	88 %
7	0.1750	92 %	5	98 %
8	0.2070	98 %	13	93 %
9	0.2900	90 %	3	88 %
10	0.2070	95 %	5	95 %
11	0.2010	97 %	1	97 %
12	0.1820	93 %	7	93 %

Table 5: Performance of kNN and FDN for datasets D13 – D24

Dataset	FDN		kNN	
	k	Performance	d_c	Performance
13	3	88 %	0.1680	87 %
14	11	95 %	0.1940	97 %
15	1	87 %	0.1650	90 %
16	9	90 %	0.2350	92 %
17	5	88 %	0.1780	93 %
18	7	92 %	0.2020	88 %
19	1	97 %	0.1770	97 %
20	15	95 %	0.2120	98 %
21	3	85 %	0.2130	88 %
22	9	97 %	0.2040	97 %
23	1	97 %	0.2010	97 %
24	5	95 %	0.1790	93 %

The optimal parameters for kNN were found in a similar manner.

When creating the feature vector datasets D1 to D12, the parameters were chosen such that they produced the best performance for FDN classification method during the training period.

On the other hand, the feature vector datasets D13 to D24 were constructed using, the optimal parameters found for kNN during the training period.

The performance of a given dataset was calculated using the formula given below.

$$P = \frac{N_s}{N} \times 100 \qquad \ldots(1)$$

P – Performance
N_s – Number of successfully identified mental tasks
N – Total number of mental tasks

The performance of FDN and kNN are shown in Tables 4 and 5. k values for kNN and d_c values for FDN given in the tables corresponding to the best performance that were obtained for each method. It is evident from Tables 4 and 5 that FDN method performed better than kNN for most of the feature vector datasets used in the study.

DISCUSSION AND CONCLUSION

The performance of a new classification method, fixed distance neighbour classifier, which is closely related to the well known k nearest neighbour classifier is presented.

The main difference between these two methods can be summarized as follows. kNN does not consider how far the training dataset points are located from the test point but considers only the number of neighbours. As a result some neighbours, which participated in voting can be far away from the test point due to lack of training data points in the neighbourhood of the test point. On the other hand FDN does consider how far the training data set points are located from the test point, but does not take into account the number of neighbours that participated in voting when classification decision was made. As a result, for different test points, different number of neighbours participated in voting. However, if the distribution of feature vectors in the feature space (i.e. density of feature vectors) is uniform, both methods should produce the same performance.

For most of the twenty four feature vector data sets used in this study, FDN showed slightly better performance than the kNN, indicating FDN is a viable classification method, which can be used in place of kNN. Since we have tested FDN only for feature vectors derived from EEG data sets, which were specifically recorded for BCI, the true merit of FDN as a general classification method cannot be made and further studies have to be carried out with other type of data to determine its general validity.

REFERENCES

1. Barreto G.A., Frota R.A. & de Medeiros F.N.S. (2004). On the classification of mental tasks: a performance comparison of neural and statistical approaches. *Proceeding of the IEEE Workshop on Machine Learning for Signal Processing,* 29 September - 4 October.

2. Birbaumer N. (1999). Slow cortical potentials: plasticity, operant control, and behavioural effects. *Neuroscientist* **5**(2):74 –78.

3. Chiappa S. & Bengio S. (2004). HMM and IOHMM modeling of EEG rhythms for asynchronous BCI systems. *European Symposium on Artificial Neural Networks (ESANN)*, April 28 − 30, Bruges, Belgium.

4. Congedo M., Lotte F. & Lécuyer A. (2006). Classification of movement intention by spatially filtered electromagnetic inverse solutions. *Physics in Medicine and Biology* **51**: 1971– 1989.

5. Domeniconi C., Peng J. & Gunopulos D. (2002). Locally adaptive metric nearest-neighbour classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(9): 1281 – 1285.

6. Kamousi B., Liu Z. & He B. (2005). Classification of motor imagery tasks for brain–computer interface applications by means of two equivalent dipoles analysis. *IEEE Transactions* on *Neural Systems and Rehabilitation Engineering* **13**: 166 – 171.

7. Kaper M., Meinicke P., Grossekathoefer U., Lingner T. & Ritter H. (2004). BCI competition 2003 – data set IIb: support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engineering* **51**: 1073 – 1076.

8. Lehtonen J., Jylänki P., Kauhanen L. & Sams M. (2008). Online classification of single EEG trials during finger movements. *IEEE Transactions on Biomedical Engineering* **55**(2):713 –720.

9. Lotte F., Congedo M., Lécuyer A., Lamarche F. & Arnaldi B. (2007). A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering* **4**: R1 – R13.

10. Mason S.G. & Birch G.E. (2000). A brain-controlled switch for asynchronous control applications. *IEEE Transactions on Biomedical Engineering* **47**: 1297–1307.

11. Millán J.R. & Mouriño J. (2003). Asynchronous BCI and local neuralclassifiers: an overview of the adaptive brain interface project. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* **11**: 159 –161.

12. Penny W.D., Roberts S.J., Curran E.A. & Stokes M.J. (2000). EEG-based communication: a pattern recognition approach. *IEEE Transactions on Rehabilitation Engineering* **8**: 214 –215.

13. Pfurtscheller G., Brunner G.C., Schlogl A. & da Silva F.H.L. (2006). Mu rhythm (de) synchronization and EEG single-trial classification of different motor imagery tasks. *Neuroimage* **3**:153 –159.

14. Pfurtscheller G., Neuper C., Flotzinger D. & Pregenzer M. (1997). EEG-based discrimination between imagination of right and left hand movement *Electroencephalography and Clinical Neurophysiology* **103**: 642 – 651.

15. Pfurtscheller G., Neuper C., Schlogl A. & Lugger K. (1998). Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters. *IEEE Transactions on Rehabilitation Engineering* **6**: 316 –325.

16. Qin L., Ding L. & He B. (2004). Motor imagery classification by means of source analysis for brain–computer interface applications. *Journal of Neural Engineering* **1**: 135 –141.

17. Voulgaris Z. & George D. Magoulas (2008). Extensions of the k nearest neighbour methods for classification problems. *Proceedings of the 26th IASTED International Conference on Artificial Intelligence and Application, AIA 2008*, pp. 23 –28.

18. Wang T., Deng J. & He B. (2004). Classifying EEG-based motor imagery tasks by means of time-frequency synthesized spatial patterns. *Clinical Neurophysiology* **115**: 2744 – 2753.

19. Zachary A.K. & Jorge I.A. (1990). A new mode of communication between man and his surroundings. *IEEE Transactions on Biomedical Engineering* **37**(12): 1209 –1241.