Journal of Computer Applications (JCA)
ISSN: 0974-1925, Volume IV, Issue 3, 2011

Comparison of Classification Methods Using EEG Data

Mrs.V.Baby Deepa[a,*], Dr. P.Thangaraj [b,1]

Abstract - Brain Computer Interface (BCI) enables the capturing and processing of motor imagery related brain signals which can be interpreted by computers. BCI systems capture the motor imagery signals via Electroencephalogram or Electrocorticogram. The processing of the signal is usually attempted by extracting feature vectors in the frequency domain and using classification algorithms to interpret the motor imagery action. In this paper we investigate the motor imagery signals obtained from dataset using the Fast Hartley Transform (FHT) for feature vector extraction and feature reduction using support vector machine. The processed data is trained and classified using the Bayes Net , Navie Bayes and Bayesian Logistic Regression.

Index Terms - *EEG (Electro-encephalogram), BCI (Brain Computer Interface), Navie Bayes Bayes Net and Bayesian Logistic Regression.*

I. INTRODUCTION

A brain-computer interface, sometimes called a direct neural interface or a brain machine interface, is a direct communication pathway between a human or animal brain(or brain cell culture) and an external device. BCIs are often aimed at assisting, augmenting or repairing human cognitive or sensory-motor functions. In one BCIs, [1,2] computers either accept commands from the brain or send signals to it but not both. Two way BCIs will allow brains and external devices to exchange information in both directions but have yet to be successfully implanted in animals or humans.

Brain-machine interfaces promise to aid paralyzed patients by re-routing movement-related signals around damaged parts of the nervous system. A new study in Nature demonstrates a human with spinal injury manipulating a screen cursor and robotic devices by thought alone. Implanted electrodes in his motor cortex recorded neural activity, and translated it into movement commands.
With recent advances in technology and knowledge, pioneering researchers could now conceivably attempt to produce BCIs that augment human functions rather than simply restoring them, previously only a possibility in science fiction.

II. EEG DATA

An **electroencephalogram** (EEG) is a test used to detect abnormalities related to electrical activity of the brain. This procedure tracks and records brain wave patterns. Small metal discs with thin wires (electrodes) are placed on the scalp, and then send signals to a computer to record the results.

Manuscript received, 2011.
Mrs.V.Baby Deepa [a,*],
Asst.Professor
M Kumarasamy College of Engineering
Karur- 639 113
E-mail:deepamct@gmail.com

Dr. P.Thangaraj [b,1],
Prof. and Head
Dept. of Computer Science & Engg.
BIT, Sathyamangalam, Erode.
Email : ctpt@yahoo.co.in

Normal electrical activity in the brain makes a recognizable pattern. Through an EEG [3], doctors can look for abnormal patterns that indicate seizures and other problems.
The most common reason an EEG [4] is performed is to diagnose and monitor seizure disorders. EEGs can also help to identify causes of other problems such as sleep disorders and changes in behavior. EEGs are sometimes used to evaluate brain activity after a severe head injury or before heart or liver transplantation.

III. DISCRETE HARTLEY TRANSFORM

A **discrete Hartley transform** (DHT) is a Fourier-related transform of discrete, periodic data similar to the discrete Fourier transform (DFT), with analogous applications in signal processing and related fields. Its main distinction from the DFT is that it transforms real inputs to real outputs, with no intrinsic involvement of complex numbers. Just as the DFT is the discrete analogue of the continuous Fourier transform, the DHT is the discrete analogue of the continuous Hartley transform. Because there are fast algorithms for the DHT analogous to the fast Fourier transform (FFT), the DHT was originally proposed by R. N. Bracewell in 1983 as a more efficient computational tool in the common case where the data are purely real. It was subsequently argued, however, that specialized FFT algorithms for real inputs or outputs can ordinarily be found with slightly fewer operations than any corresponding algorithm for the DHT. In particular, the DHT analogue of the Cooley-Tukey algorithm is commonly known as the **Fast Hartley Transform** (FHT) [5] algorithm.

Discrete Hartley transform is an analogue of discrete Fourier transform for real data. The Hartley transform takes a real sequence as an input. The result is also a real sequence:

$$H_k = \sum_{n=0}^{N-1} x_n \cdot \left(\cos\left(\frac{2\pi nk}{N}\right) + \sin\left(\frac{2\pi nk}{N}\right)\right)$$

For some time it was considered that Hartley transform can be a faster alternative to the real Fourier transform, but later it was found out that there are FFT algorithms, which are a little more efficient than the corresponding FHT [6, 7] algorithms.

IV SUPPORT VECTOR MACHINE

A **support vector machine** (**SVM**) [8,9] is a concept in computer science for a set of related supervised learning methods that analyze data and recognize patterns, used for classification and regression analysis. The standard SVM takes a set of input data and predicts, for each given input, which of two possible classes the input is a member of, which makes the SVM a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of

the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite- dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

The original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mapping used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $K(x,y)$ selected to suit the problem. The hyperplanes in the higher dimensional space are defined as the set of points whose inner product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters α_i of images of feature vectors that occur in the data base. With this choice of a hyperplane, the points x in the feature space that are mapped into the hyperplane are defined by the relation:

$$\sum \alpha_i K(x_i,x) = constant$$

Note that if $K(x,y)$ becomes small as y grows further from x, each element in the sum measures the degree of closeness of the test point x to the corresponding data base point x_i. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyperplane can be quite convoluted as a result allowing much more complex discrimination between sets which are not convex at all in the original space.

V BAYESIAN NETWORKS

Bayesian networks (BNs), also known as *belief networks* (or Bayes nets for short), belong to the family of probabilistic *graphical models* (GMs). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. Hence, BNs combine principles from graph theory, **probability theory**, computer science, and statistics.

GMs with *undirected edges* are generally called *Markov random fields* or *Markov networks*. These networks provide a simple definition of independence between any two distinct nodes based on the concept of a *Markov blanket*. Markov networks are popular in fields such as statistical physics and computer vision.

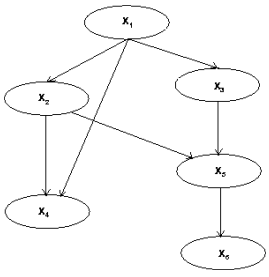
BNs correspond to another GM structure known as a *directed acyclic graph* (DAG) that is popular in the statistics, the machine learning, and the artificial intelligence societies. BNs are both mathematically rigorous and intuitively understandable. They enable an effective representation and computation of the joint probability distribution (JPD) over a set of random variables.

The structure of a DAG is defined by two sets: the set of nodes (vertices) and the set of directed edges. The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependence among the variables and are drawn by arrows between nodes. In particular, an edge from node *Xi* to node *Xj* represents a statistical dependence between the corresponding variables. Thus, the arrow indicates that a value taken by Variable *Xj* depends on the value taken by variable *Xi,* or roughly speaking that variable *Xi* "influences" *Xj* . Node *Xi* is then referred to as a *parent* of *Xj* and, similarly, *Xj* is referred to as the *child* of *Xi.* An extension of these genealogical terms is often used to define the sets of "descendants" – the set of nodes that can be reached on a direct path from the node, or "ancestor" nodes – the set of nodes from which the node can be reached on a direct path. The structure of the acyclic graph guarantees that there is no node that can be its own ancestor or its own descendent. Such a condition is of vital importance to the factorization of the joint probability of a collection of nodes as seen below. Note that although the arrows represent direct causal connection between the variables, the *reasoning process* can operate on BNs by propagating information in any direction .

A BN reflects a simple conditional independence statement. Namely that each variable is independent of its nondescendents in the graph given the state of its parents. This property is used to reduce, sometimes significantly, the number of parameters that are required to characterize the JPD of the variables. This reduction provides an efficient way to compute the posterior probabilities given the evidence.

In addition to the DAG structure, which is often considered as the "qualitative" part of the model, one needs to specify the "quantitative" parameters of the model. The parameters are described in a manner which is consistent with a Markovian property, where the conditional probability distribution (CPD) at each node depends only on its parents. For discrete random variables, this conditional probability is often represented by a table, listing the local probability that a child node takes on each of the feasible values – for each combination of values of its parents. The joint distribution of a collection of variables can be determined uniquely by these local conditional probability tables (CPTs).

Bayesian networks are used to represent essential information in databases in a network structure. The network consists of edges $P(x_6 \mid x_5)P(x_5 \mid x_2,x_3)P(x_4 \mid x_2,x_1)P(x_3 \mid x_1)P(x_2 \mid x_1)P(x_1)$ and vertices, where the vertices are *events* and the edges *relations* between events. A simple Bayesian network is illustrated in figure where symptoms are dependent on a disease, and a disease is dependent on age, work and work environment. Bayesian networks are easy to interpret for humans, and are able to store *causal relationships*, that is, relations between causes and effects. The networks can be used to represent domain knowledge, and it is possible to control inference and produce explanations on a network. A Bayesian network is shown in Fig, representing the probability distribution P:



A Bayesian Network Representing the Distribution P.

VI NAIVE BAYES CLASSIFIER

Bayes classifier [10] is a simple probabilistic classifier based on applying Bayes' theorem (from Bayesian statistics) with strong (naive) independence assumptions. A more descriptive term for the underlying probability model would be "independent feature model".

Depending on the precise nature of the probability model, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without believing in Bayesian probability or using any Bayesian methods.

VII LOGISTIC REGRESSION

Logistic regression (sometimes called the **logistic model** or **logit model**) [11] is used for prediction of the probability of occurrence of an event by fitting data to a logit function logistic curve. It is a generalized linear model used for binomial regression. Like many forms of regression analysis, it makes use of several predictor variables that may be either numerical or categorical.

VIII RESULT

Bayes Net result

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	61.3095 %
Incorrectly Classified Instances	38.6905 %
Kappa statistic	0.2178
Mean absolute error	0.4364
Root mean squared error	0.497
Relative absolute error	87.4669 %
Root relative squared error	99.5084 %
Total Number of Instances	168

Detailed Accuracy By Class

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.5	0.28	0.615	0.5	0.552	0.631	hand
0.72	0.5	0.612	0.716	0.66	0.631	foot

Weighted Avg.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
0.61	0.397	0.613	0.613	0.608	0.631

=== Confusion Matrix ===

a	b	<-- classified as
40	40	a = hand
25	63	b = foot

Naïve bayes

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	57.1429 %
Incorrectly Classified Instances	42.8571 %
Kappa statistic	0.1419
Mean absolute error	0.4357
Root mean squared error	0.6462
Relative absolute error	87.338 %
Root relative squared error	129.3775 %
Total Number of Instances	168

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.56	0.42	0.549	0.563	0.556	0.548	hand
0.58	0.44	0.593	0.58	0.586	0.552	foot

Weighted Avg.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
0.57	0.429	0.572	0.571	0.572	0.55

=== Confusion Matrix ===

a	b	<-- classified as
45	35	a = hand
37	51	b = foot

Bayesian Logistic regression

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances	46.4286 %
Incorrectly Classified Instances	53.5714 %
Kappa statistic	-0.085
Mean absolute error	0.5357
Root mean squared error	0.7319
Relative absolute error	107.3768 %
Root relative squared error	146.5425 %
Total Number of Instances	168

Detailed Accuracy By Class

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.33	0.41	0.419	0.325	0.366	0.458	hand
0.59	0.68	0.491	0.591	0.536	0.458	foot

Weighted Avg.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
0.46	0.55	0.457	0.464	0.455	0.458

=== Confusion Matrix ===

a	b	<-- classified as
26	54	a = hand
36	52	b = foot

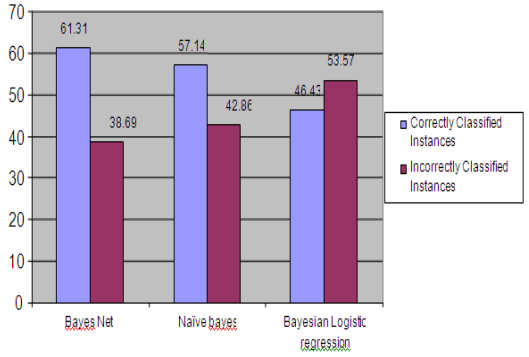


Figure 1: CLASSIFICATION ACCURACY

IX CONCLUSION

In this paper feature vector was extracted from the dataset using Fast Hartley Transform. Sub set selection of the obtained features after normalization was achieved using Support Vector Machine. Bayes Net**,** Naïve bayes, logistic regression was used to train and classify the extracted sub features. Result show classification accuracy of over 60% in Bayes Net. Further investigation has to be done to improve the classification accuracy on a small number of attributes.

REFERENCES

1. E.E. Sutter, The brain response interface: communication through visually induce delectrical brain responses 1992, Journal of Microcomputer Applications, v. 15, pp. 31-45.
2. Chang S. Nam, Gerwin Schalk, Melody Moore Jackson: Current Trends in Brain-Computer Interface (BCI) Research and Development. Int. J. Human. Computer. Interaction 27(1): 1-4 (2011)
3. Michael Adamaszek, Sebastian Olbrich and Jürgen Gallinat, The Diagnostic Value of Clinical EEG in Detecting Abnormal Synchronicity in Panic Disorder Journal of Clinical EEG & Neuroscience, July, 2011.
4. Eickhoff SB, Grefkes C Approaches for the integrated analysis of structure, function and connectivity of the human brain. [Journal Article, Research Support, N.I.H., Extramural, Research Support, Non-U.S.Gov't] Clin EEG Neurosci 2011 Apr; 42(2):107-21.
5. R. N. Bracewell, "The fast Hartley transform," Proc. IEEE, vol. 72, no. 8, pp. 1010-1018, Aug. 1984.
6. H.J.Meckelburg and D.Lipka, "Fast Hartley transform algorithm," Electronics Letters, vol. 21, no. 8, pp. 311-313, Apr. 1985.
7. Bracewell, R. N. *The Hartley Transform.* New York: Oxford University Press, 1986.
8. T.Joachims, SVM Light Support Vector Machine, 2002, <http://svmlight.joachims.org>.
9. D.Koller, M. Sahami, Hierarchically classifying documents using very few words, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 170–178.
10. Effective Anomaly based Intrusion Detection using Rough Set Theory and Support Vector Machine International Journal of Computer Applications © 2011 by IJCA Journal Number 3 - Article 8 Year of Publication: 2011
11. Caruana, R. and Niculescu-Mizil, A.: "An empirical comparison of supervised learning algorithms". Proceedings of the 23rd international
12. conference on Machine learning, 2006.
13. Hilbe, Joseph M. (2009). Logistic Regression Models. Chapman & Hall/CRC Press. ISBN 978-1-4200-7575-5.

BIOGRAPHY

V.Baby Deepa, received her Bachelor's and Master's in Computer Science from Barathidasan University, Trichy and did her M.Phil. as well in the same university. She has 12 years of teaching experience and is Assistant professor in the faculty of Software Engineering, she is serving as the head for the same faculty in M.Kumarasamy College of Engineering, Karur. She has presented more than 15 papers on various topics including national, international conference and journals. She is a research scholar of Anna University Chennai and her research area is Fuzzy and Data Mining.

Dr.P.Thangaraj, did his graduation and post graduation in Mathematics at Madras University.He completed his M.Phil degree in the year 1993 from Bharathiar University. He completed his research work on Fuzzy Metric Spaces and awarded Ph.D degree by Bharathiar University.He completed the post graduation in Computer Applications at ICNOU in 2005. He completed his Master of Engineering degree in Computer Science in the year 2007 at Vinayaka Missions University. Currently he is a professor and Head of Computer Science and Engineering in Bannariamman College of Engineering and Technology. His current area of research interest is in Fuzzy Metric Spaces and Data Mining.