Classification of EEG data using FHT and SVM based on Bayesian Network

V. Baby Deepa[1] and Dr. P. Thangaraj[2]
[1]Dept. of Software Engnineering, M.Kumarasamy College of Engineering
Karur, Tamil Nadu, 639 113, India

[2]Dept. of Computer Science and Engineering, Bannari Amman Institute of Technology
Sathyamangalam, Erode, Tamil Nadu, 638 401, India

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 BCI competition dataset IVA 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.

Keywords: *Brain Computer Interface (BCI),Electroencephalogram (EEG), Fast Hartley Transform (FHT), Bayes Net (BN)*

1. Introduction

A Brain Computer Interface [1], also known as 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 focused on assisting, augmenting or repairing human cognitive or sensory-motor functions. BCIs would act in two ways. In the case of one-way BCIs, computers either accept commands from the brain or send signals to it but not both. Two way BCIs allow brains and external devices to exchange information in both directions but it is not yet successful in the aspect of implantation.

Brain-machine interfaces [2] help paralyzed patients by re-routing movement-related signals around damaged parts of the nervous system. With recent advancement in technology and knowledge, the researchers now conceivably attempt to produce BCIs that augment human functions rather than simply restoring them.

In this paper, the motor imagery signals obtained from BCI competition dataset IVA are investigated. They are further exposed to Fast Hartley Transform and Support Vector Machine for feature vector extraction and feature reduction respectively. In this paper, Section I gives an introduction to BCI, Section II describes the dataset and EEG data, Section III and IV explain the feature vector extraction and feature reduction using FHT and SVM, Section V briefs about the classification using Bayes Net and is followed by the experimental results and conclusion.

2. Data Set and EEG

The IV A dataset used in the brain computer interface competition provided by Intelligent Data Analysis Group has been taken for investigation. It consists of recordings from five healthy subjects who sat in a chair with arms resting on armrests. Visual cues indicated for 3.5 s which of the following 3 motor imageries the subject should perform: (L) *left* hand, (R) *right* hand, (F) right *foot*. The presentation of target cues was intermitted by periods of random length, 1.75 to 2.25 s, in which the subject could relax. Given are continuous signals of 118 EEG channels and markers that indicate the time points of 280 cues for each of the 5 subjects (*aa*, *al*, *av*, *aw*, *ay*). Subject aa was used in our study.

The abnormalities related to electrical activity of the brain could be detected using a test called electroencephalogram (EEG) [3] which tracks and records brain wave patterns. Electrodes (Small metal discs with thin wires) are placed on the scalp, which send signals to a computer to record the results. A recognizable pattern of EEG is identified as a normal electrical activity in the brain. EEG is used to detect abnormal patterns of brains that indicate seizures and other problems. EEG is also performed to diagnose monitor seizure disorders, sleep disorders and other changes in behavior. It is also used to evaluate brain activity after a severe head injury or before heart or liver transplantation.

3. Feature Vector Extraction

In this paper the feature vector extraction is performed on the dataset using Fast Hartley Transform (FHT). FHT [4] is a technique helps to extract the feature vector efficiently. 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. As the DFT is the discrete analogue of the continuous Fourier transform, the DHT is the discrete analogue of the continuous Hartley transform.

Since there are fast algorithms for the DHT analogous to the fast Fourier transform (FFT), the DHT was proposed 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. 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:

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

It was considered, for sometime, 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] algorithms. An integral transform which shares some features with the Fourier transform, but which (in the discrete case), multiplies the integral kernel by

$$\cos\left(\frac{2\pi k n}{N}\right) - \sin\left(\frac{2\pi k n}{N}\right) \quad (1)$$

instead of

$$e^{-2\pi i k n/N} = \cos\left(\frac{2\pi k n}{N}\right) - i \sin\left(\frac{2\pi k n}{N}\right). \quad (2)$$

The Hartley transform produces real output for a real input, and is its own inverse. It therefore can have computational advantages over the discrete Fourier transform, although analytic expressions are usually more complicated for the Hartley transform.

The discrete version of the Hartley transform can be written explicitly as

$$\mathcal{H}[a] \equiv \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} a_n \left[\cos\left(\frac{2\pi k n}{N}\right) - \sin\left(\frac{2\pi k n}{N}\right)\right] \quad (3)$$

$$= \Re \mathcal{F}[a] - \Im \mathcal{F}[a], \quad (4)$$

where $\mathcal{F}$ denotes the Fourier transform. The Hartley transform obeys the convolution property

$$\mathcal{H}[a * b]_k = \tfrac{1}{2}\left(A_k B_k - \overline{A}_k \overline{B}_k + A_k \overline{B}_k + \overline{A}_k B_k\right), \quad (5)$$

where

$$\overline{a}_0 \equiv a_0 \quad (6)$$

$$\overline{a}_{n/2} \equiv a_{n/2} \quad (7)$$

$$\overline{a}_k \equiv a_{n-k}. \quad (8)$$

Like the fast Fourier transforms, there is a fast version of the Hartley transform. Decimation in time algorithm makes use of

$$\mathcal{H}_n^{\text{left}}[a] = \mathcal{H}_{n/2}[a^{\text{even}}] + \mathcal{X}\,\mathcal{H}_{n/2}\left[a^{\text{odd}}\right] \quad (9)$$

$$\mathcal{H}_n^{\text{right}}[a] = \mathcal{H}_{n/2}[a^{\text{even}}] - \mathcal{X}\,\mathcal{H}_{n/2}\left[a^{\text{odd}}\right], \quad (10)$$

where $\mathcal{X}$ denotes the sequence with elements

$$a_n \cos\left(\frac{\pi n}{N}\right) - \overline{a}_n \sin\left(\frac{\pi n}{N}\right). \quad (11)$$

Decimation in frequency algorithm makes use of

$$\mathcal{H}_n^{\text{even}}[a] = \mathcal{H}_{n/2}\left[a^{\text{left}} + a^{\text{right}}\right] \quad (12)$$

$$\mathcal{H}_n^{\text{odd}}[a] = \mathcal{H}_{n/2}\left[\mathcal{X}\left(a^{\text{left}} - a^{\text{right}}\right)\right]. \quad (13)$$

The discrete Fourier transform

$$A_k \equiv \mathcal{F}\,[a] = \sum_{n=0}^{N-1} e^{-2\pi i k n/N}\, a_n \qquad (14)$$

can be written

$$\begin{bmatrix} A_k \\ A_{-k} \end{bmatrix} \qquad \sum_{n=0}^{N-1} \underbrace{\begin{bmatrix} e^{-2\pi i k n/N} & 0 \\ 0 & e^{2\pi i k n/N} \end{bmatrix}}_{\mathsf{F}} \begin{bmatrix} a_n \\ a_n \end{bmatrix}$$

So

$$= \sum_{n=0}^{N-1} \underbrace{\frac{1}{2}\begin{bmatrix} 1-i & 1+i \\ 1+i & 1-i \end{bmatrix}}_{\mathsf{T}^{-1}} \underbrace{\begin{bmatrix} \cos\left(\frac{2\pi k n}{N}\right) & \sin\left(\frac{2\pi k n}{N}\right) \\ -\sin\left(\frac{2\pi k n}{N}\right) & \cos\left(\frac{2\pi k n}{N}\right) \end{bmatrix}}_{\mathsf{H}} \underbrace{\frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}}_{\mathsf{T}} \begin{bmatrix} a_n \\ a_n \end{bmatrix},$$

$$\mathsf{F} = \mathsf{T}^{-1}\,\mathsf{H}\,\mathsf{T}. \qquad (15)$$

4. Feature Reduction

A support vector machine (SVM) [7] 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 [8] 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 [9] 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:

$$\sum_i \alpha_i K(x_i, x) = \mathrm{constant}$$

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. 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.

5. Classification Using Bayes Net

Bayesian networks (BNs) [10], 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, Bayesian Networks combine principles from graph theory, probability theory, computer science, and statistics.

Graphical Models 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 [11] 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 [12] reflects a simple conditional independence statement. Namely that each variable is independent of its non descendents 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 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 simple usage of Bayesian networks is denoted naive Bayesian classification. These networks consist only of one parent and several child nodes. Classification is done by considering the parent node to be a hidden variable (H in the figure) stating which class (child node) each object in the database should belong to. An existing system using naive Bayesian classification is AutoClass.

The theoretical foundation for Bayesian networks is Bayes rule, which states:

$$P(H \mid e) = \frac{P(e \mid H)P(H)}{P(e)}$$

where *H* is a hypothesis, and *e* an event. $P(e \mid H)$ is the posterior probability, and *P*(*H*) is the prior probability. To give a formal definition of Bayesian networks, we introduce some terminology which is taken from:

If a subset of *Z* nodes in a graph *G* intercepts all paths between the nodes *X* and *Y* (written), then this $\langle X \mid Y \mid Z \rangle_G$ corresponds to *conditional independence* between *X* and *Y* given *Z*:

$$\langle X \mid Y \mid Z \rangle_G \Rightarrow I(X, Z, Y)_M$$

conversely:

$$I(X, Z, Y)_M \Rightarrow \langle X \mid Y \mid Z \rangle_G$$

with respect to some dependency model M.

A Directed, Acyclic Graph (DAG) *D* is said to be a *I-map* of a dependency model *M* if for every three disjoint sets of vertices, *X*, *Y* and *Z* we have:

$$\langle X \mid Y \mid Z \rangle_G \Rightarrow I(X, Z, Y)_M$$

A DAG is a minimal I-map of *M* if none of its arrows can be deleted without destroying its I-mapness. Given a probability distribution *P* on a set of variables *U*, a DAG is called a *Bayesian Network* of P if $D = (U, \vec{E})$ and only if *D* is a minimal I-map of *P*.

A Bayesian network is shown in Fig, representing the probability distribution P:

$$P(x_6 \mid x_5)P(x_5 \mid x_2, x_3)P(x_4 \mid x_2, x_1)P(x_3 \mid x_1)P(x_2 \mid x_1)P(x_1)$$

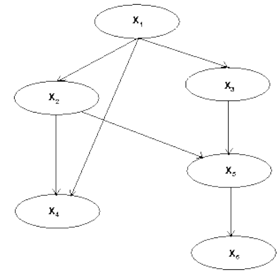

Fig. 1: A Bayesian Network Representing the Distribution P.

6. Experimental Result

Bayes Net

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

Correctly Classified Instances	103	61.3095 %
Incorrectly Classified Instances	65	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 ===

	TPRate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.5	0.284	0.615	0.5	0.552	0.631	hand
	0.716	0.5	0.612	0.716	0.66	0.631	foot
Weighted Avg.	0.613	0.397	0.613	0.613	0.608	0.631	

=== Confusion Matrix ===

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

7. Conclusion

In this paper feature vector was extracted from the BCI competition IVA dataset using Fast Hartley Transform. Sub set selection of the obtained features after normalization was achieved using Support Vector Machine. Decision tree and logistic regression based on Bayes probability was used to train and classify the extracted sub features. Results show that the classification accuracy is over 60% in Bayesian Network. 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 induced electrical 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]. R. N. Bracewell, “The fast Hartley transform,” Proc. IEEE, vol. 72, no. 8, pp. 1010-1018, Aug. 1984.

[5]. H. J. Meckelburg and D. Lipka, “Fast Hartley transform algorithm,” Electronics Letters, vol. 21, no. 8, pp. 311-313, Apr. 1985.

[6]. Bracewell, R. N. *The Hartley Transform.* New York: Oxford University Press, 1986.

[7]. T. Joachims, SVM Light Support Vector Machine, 2002, 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.

[8]. Shailendra Kumar, Shrivastava Preeti Jain Effective Anomaly based Intrusion Detection using Rough Set Theory and Support Vector Machine, International Journal of Computer Applications,IJCA, Journal Number 3 - Article 8 Year of Publication: 2011, doi: 10.5120/2261-2906

[9]. Hui Xue; Songcan Chen; Qiang Yang; Structural Regularized Support Vector Machine: A Framework for Structural Large Margin Classifier, Neural Networks, IEEE Transactions on , Issue Date: April 2011, Volume: 22 Issue:4, On page(s): 573 – 587, ISSN: 1045-9227

[10]. Pearl, J. & Russel, S. (2001). Bayesian networks. Report (R-277), November 2000, in Handbook of BrainTheory and Neural Networks, M. Arbib, ed, MIT Press, Cambridge, pp. 157–160.

[11]. Friedman, N. & Goldszmidt, M. (1996). Learning Bayesian networks with local structure, in Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence, Portland, August 1–4 1996.

[12]. Boutilier, C., Friedman, N., Goldszmidt, M. & Koller, D. (1996). Context-specific independence in Bayesian networks, in Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence, Portland, August 1–4 1996, pp. 115–123.

V.Baby Deepa, received her Bachelors and Masters degree in Computer Science from Barathidasan University, Trichy and received her M.Phil. Degree as well from the same university. She has 14 years of teaching experience. Besides being an 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 16 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, received Bachelor's and Master's in Mathematics from Madras University in 1981 and 1983. He completed his M.Phil degree in the year 1993 from Bharathiar University and research work on Fuzzy Metric Spaces and awarded Ph.D degree by Bharathiar University. He completed the post graduation in Computer Applications at IGNOU in 2005 and Master of Engineering degree in Computer Science in the year 2007 at Vinayaka Missions University. Currently he is a Professor Head of Computer Science and Engineering in Bannariamman College of Engineering and Technology.