Analyzing EEG data from the Brain Computer Interface with Independent Component Analysis

Rachel Blake mentored by Dr. Arye Nehorai

Preston M. Green Department of Electrical and Systems Engineering, Washington University in St. Louis

Introduction

Independent Component Analysis (ICA) is a form of feature extraction and blind source separation. Feature extraction is a step in pattern classification that creates a new feature subset based on transformations and combinations of the original feature set [1]. The goal of ICA for feature extraction is to find the most linearly independent signals possible based on the original signal. Blind source separation is a method for calculating an original signal based on a mixed signal and with no knowledge of the mixing process or the original signal.

The concept of ICA can be explained through an adaptation of the cocktail party problem. If there are two conversations going on in opposite corners of a room and there are two recording devices, the recordings will be a mixture of the two conversations. The goal of ICA is to separate the original conversations out of the mixed signals [2].

BCI Background and Data

The Brain Computer Interface (BCI) is a way of using brain waves to send commands to a computer. It has important application for handicapped individuals and individuals that have suffered from brain injury who no longer have full physical capabilities. The goal of this project is to increase the accuracy of the signals processing stage of the BCI.

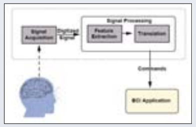

Figure 1: The flow of the signal in a BCI.

Data Set

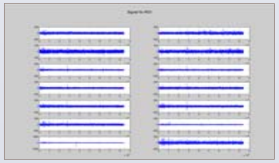

Figure 2: Original 14 EEG signals.

The data set used for this project was EEG data collected from 14 nodes on a subjects head and this makes the feature space have a dimension of 14. In each trial, the subject was asked to imagine right hand movement at specific times and this stimulus was recorded along with the signal recorded at each node. The features being considered are the signals from the different nodes.

ICA Algorithm

Goal is to maximize the non-gaussianity which is a measure of independence.

1. Choose an initial (e.g. random) weight vector **w**.
2. Let $\mathbf{w}+ = E\{\mathbf{x}g(\mathbf{w}T\,\mathbf{x})\} - E\{g0(\mathbf{w}T\,\mathbf{x})\}\mathbf{w}$
3. Let $\mathbf{w} = \mathbf{w}+/||\mathbf{w}+||$
4. If not converged, go back to 2.

This is an algorithm for extracting only one independent component [2].

Reduction of Dimensionality

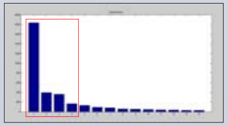

Figure 3: Eigenvalues used for reducing dimensionality

ICA is used as a method to reduce the dimensionality of a feature space. By selecting the eigenvalues with highest value one can select the most salient features.

Classifiers

Decision Tree

A decision tree is recursively created by selecting the split in the data that will lead to the purest sub-datasets in the children nodes. Test samples are then classified by comparing the selected feature at each node to the split values until a leaf node is reached [3].

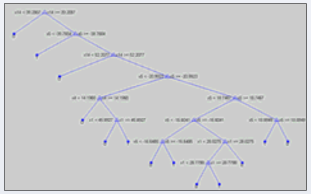

Figure 4: Part of decision tree used in classifying ICA signals

Linear Discriminant Analysis (LDA)

LDA classification finds a linear combination of features which separates two classes. Test samples are then classified by mapping them to the class boundary and classifying based on a selected or calculated threshold [4].

I selected my threshold value by looking at the LDA classifier for the ICA signal with 14 dimensions and plotting the probability of classifying each class incorrectly at different thresholds. I then selected a threshold that performed well for both but minimized the risk of classifying imagined right hand movement wrong.

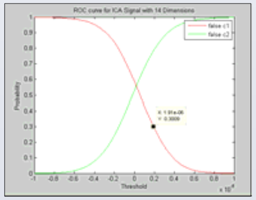

Figure 5: ROC curve comparing the probability of false classification of right hand movement and the probability of false classification of another stimulus.

Naïve Bayes

A Naïve Bayes classifier calculates and maximizes the posteriori probability of a test sample being in a specific class with the following formula. The training data are used to calculate the prior probabilities [5]:

$$c = \underset{c_j}{\arg\max}\, \Pr(c_j) \prod_{i=1}^{|A|} \Pr(A_i = a_i \mid C = c_j)$$

This formulas is derived with Bayes' Theorem.

- A_i = attributes, a_i = observed values of attributes
- C = the actual class, c_j = the predicted class

Applications of Classifiers

Classifiers have applications in many different fields. They can be used for classifying images, sounds, language, handwriting, and human features. It has become an increasingly studied field recently due to higher levels of computing power and the widespread availability of large sets of data which can be more easily manipulated with classifiers [1].

Problem Domain	Application	Input Pattern	Pattern Classes
Bioinformatics	Sequence analysis	DNA/Protein sequence	Known types of genes/ patterns
Data mining	Searching for meaningful patterns	Points in multi-dimensional space	Compact and well-separated clusters
Document classification	Internet search	Text document	Semantic categories (e.g., business, sports, etc.)
Document image analysis	Reading machine for the blind	Document image	Alphanumeric characters, words
Industrial automation	Printed circuit board inspection	Intensity or range image	Defective / non-defective nature of product
Multimedia database retrieval	Internet search	Video clip	Video genres (e.g., action, dialogue, etc.)
Biometric recognition	Personal identification	Face, iris, fingerprint	Authorized users for access control
Remote sensing	Forecasting crop yield	Multispectral image	Land use categories, growth pattern of crops
Speech recognition	Telephone directory enquiry without operator assistance	Speech waveform	Spoken words

Table 1: Possible applications of classifiers [1]

Technique for Classifying

10-Fold Cross Validation [6]

1. Split data set into 10 groups
2. Perform classification using each group as the test set and the rest of the data as the training set
3. Average the accuracy across all of the folds

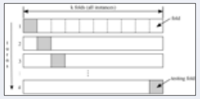

Figure 6: Example of 10-fold cross validation

Results

Decision Tree Classifier

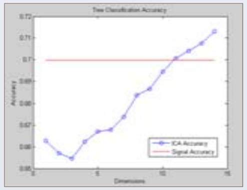

Figure 7: Accuracy of decision tree classifier for different levels of dimensionality reduction.

Reducing dimension negatively impacted accuracy of classification.

Problem: Decisions trees are sub-optimal classifiers

Linear Discriminant Analysis Classifier

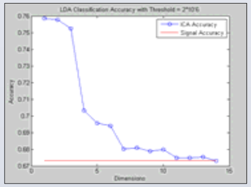

Figure 8: Accuracy of LDA classifier for different levels of dimensionality reduction.

Reducing dimension positively impacted accuracy of classification.

Problem: Results vary greatly depending on chosen threshold for classification

Naïve Bayes Classifier

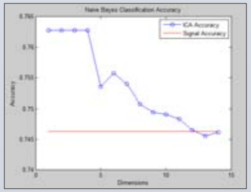

Figure 9: Accuracy of Naïve Bayes classifier for different levels of dimensionality reduction.

Reducing dimension positively impacted accuracy of classification.

Naïve Bayes Classifier is a generally reliable and efficient classifier if the conditional independence assumption is not broken.

Accuracy calculated as:

$$accuracy = \frac{number\ of\ correct\ classifications\ in\ test\ set}{total\ size\ of\ test\ set}$$

Conclusions

Based on the decision tree classifier it would seem that ICA improves the accuracy of classification when the dimensionality is not reduced and worsens the accuracy when the dimensionality is reduced. However, decision tree classifiers are often suboptimal classifiers. Based on the LDA and Naïve Bayes classifiers, reducing the dimensionality increases the accuracy of classification. These are generally more reliable classifiers so I would accept these results over those from the decision tree. However, this is not extensive enough research to conclude without hesitation that ICA improves the predictability of the stimulus.

Future Directions

Future areas of research for this topic could include:

- Other criterion for classifier error rate and accuracy
- ICA under differing conditions like using a different function to estimate the negentropy
- Other forms of blind source separation (i.e. Principal Component Analysis, etc.)
- Measure effectiveness of ICA using other, more robust classifiers
- Consider efficiency as well as accuracy in measuring the effectiveness of IC A
- Implement ICA in an online scenario

References

[1] Jain, A.K.; Duin, R. P W; Jianchang Mao, "Statistical pattern recognition: a review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.22, no.1, pp.4,37, Jan 2000

[2] A. Hyvärinen; E. Oja, "Independent component analysis: algorithms and applications", Neural Networks, Volume 13, Issues 4–5, June 2000, Pages 411-430

[3] Sreerama K. Murthy. 1998. "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey". *Data Min. Knowl. Discov.* 2, 4 (December 1998), 345-389.

[4] Sayad, S. (2010). Linear Discriminant Analysis. Retrieved April 16, 2013, from An Introduction to Data Mining website: http://www.saedsayad.com/data_mining_map.htm

[5] Mitchell, T. M. (1997). Generative and discriminative classifiers: Naive bayesand logistic regression. In *Machine Learning*. New York, NY, USA: McGraw-Hill.

[6] Borovicka, T., Jirina, M., Jr., Kordik, P., & Jirina, M. (2012). Selecting representative data sets. In A. Karahoca (Ed.), *Advances in data mining knowledge discovery and applications*.

Acknowledgments

Thank you to:

Dr. Arye Nehorai — Ed Richter

Chuck Holmes — Xiaoxiao Xu

NSF Grant No. CCF-1014908

Contact Information

For more information about this project please contact:

Rachel Blake – RachelcBlake@wustl.edu