

Emotion Classification from EEG

Rishi Dua and TV Ashok

November 20, 2014

EEL781 Neural Networks

Instructor: Dr. Jayadeva

Teaching Assistant: Sumit Soman

1 Introduction

Electroencephalogram (EEG) measures response of brain to sensory information. This information can be used to classify emotion. Since the number of features is extremely large, it is not only difficult to train a model, it might be susceptible to noise. The key idea to feature selection is that good features should be highly dependent on the labels. The primary steps involved in the analysis of are:

1. Pre-processing: The process of removing unwanted data from the dataset before using it.
2. Train-test split: We separate the dataset into train and test data using a deterministic pseudo-random split.
3. Feature Selection: Selecting relevant subset of attributes for training.
4. Benchmarking: Predicting accuracy using classifiers for each feature selection algorithm.

2 Dataset Description

DEAP, A Database for Emotion Analysis using Physiological Signals [3] consists of physiological recordings where 32 volunteers watched a set of 40

music videos and rated the videos. The data is down-sampled to 128Hz, pre-processed and segmented to 40 channels giving us 32 files, one for each user. Each file has 40 (videos) x 40 (channel) x 8064 (samples) feature matrix and a 40 (video) x 4 (label) output matrix.

3 Pre-processing

The publicly available DEAP dataset is a processed version of the original data with Noise removal and Band pass filtering done. We plotted the correlation between the features and assumed that the labels (valence, arousal, dominance and liking) are independent of each other and each label is learned and predicted separately. Due to limited computing resources, we have down-sampled the data. Instead of 32 users, we take data only for 4 of them and instead of 8064 samples, we have run experiments by using 252 readings.

4 Train-test split

We use a Pseudo-random number generator state for random sampling to make the results replicable. The seed for the sequence generator can be changed in the project configuration file.

5 Feature Selection

We use BAHSIC algorithm to select 5040 best features. To compare the performance, we generate the same number of features using Recht and Rahimi Random Fourier Features and by down sampling the features to 5040.

- **BAHSIC**

BAHSIC framework is based on dependence maximization between features and labels using Hilbert-Schmidt Independence Criterion (HSIC). HSIC tests whether random variables X and Y are independent based on a sample of observed pairs (x,y) using the Hilbert-Schmidt norm of the covariance operator between Reproducing Kernel Hilbert Space mappings of X and Y . HSIC values indicate the dependence between data and class labels serving as a criteria for Backward Elimination of features. We iteratively remove the features according to HSIC until

the desired number of features is collected. The pseudo code [4] of the algorithm is given below:

- **Recht and Rahimi's Random Fourier Features**

Expand Features as vector of $n_{randomfeatures}$:

$$\sqrt{2/n_{randomfeatures}} * \cos(X * W + B)$$

where

$$\sigma = \sqrt{2\gamma}$$

$$W = \text{normrnd}(0, \sigma, n_{features}, n_{randomfeatures})$$

$$b = 2 * \pi * \text{rand}(1, n_{randomfeatures})$$

$$B_{n_{data} \times 1} = \begin{bmatrix} b \\ b \\ \dots \\ \dots \end{bmatrix}$$

- **Down-sampling**

We select every alternate feature to keep the number of features same as in BAHSIC and RRT.

6 Benchmarking

We test the performance of the feature selection algorithms using the following classifiers. Since the ratings are from 1-9, a threshold of 4.5 is used to do a binary classification in all cases and accuracy is determined as the percentage of correctly classified test samples.

- **Linear Regression** We calculate coefficients for Ordinary least squares Linear Regression to use as a baseline.
- **Logistic Regression** We fit an L2 regularized logistic regression Coefficient of the features in the decision function.

- **Bayesian ridge regression** We fit a Bayesian ridge model and optimize the regularization parameters lambda (precision of the weights) and alpha (precision of the noise).
- **Support Vector Machine** We train epsilon-Support Vector Regression on a libsvm based implementation with $C=0.1$ and $\epsilon=0.1$. Due to resource constraints and running time, optimal parameters have not been determined using cross-validation.
- **Decision Tree** We train a decision tree regressor using a mean squared error and choosing the best split at each step.

7 Results

The accuracies of the predicted values and the actual values for each feature selection algorithm for different benchmarking techniques are given in the tables.

Benchmark	Valence	Arousal	Dominance	Liking
Linear	66.04 %	77.36%	73.58%	84.91%
Bayesian	67.92 %	77.36 %	75.47 %	100.00%
Dtree	58.49 %	83.02 %	83.02 %	88.68 %
SVR	100.00%	100.00%	100.00%	100.00%
Logistic	56.60 %	56.60 %	69.81 %	73.58%

Table 1: Accuracy using BAHSIC

Benchmark	Valence	Arousal	Dominance	Liking
Linear	43.40 %	47.17 %	56.60%	49.06%
Bayesian	100.00%	100.00%	100.00%	100.00%
Dtree	64.15 %	83.02 %	86.79 %	88.68%
SVR	100.00%	100.00%	100.00%	100.00%
Logistic	56.60 %	58.49 %	54.72 %	62.26%

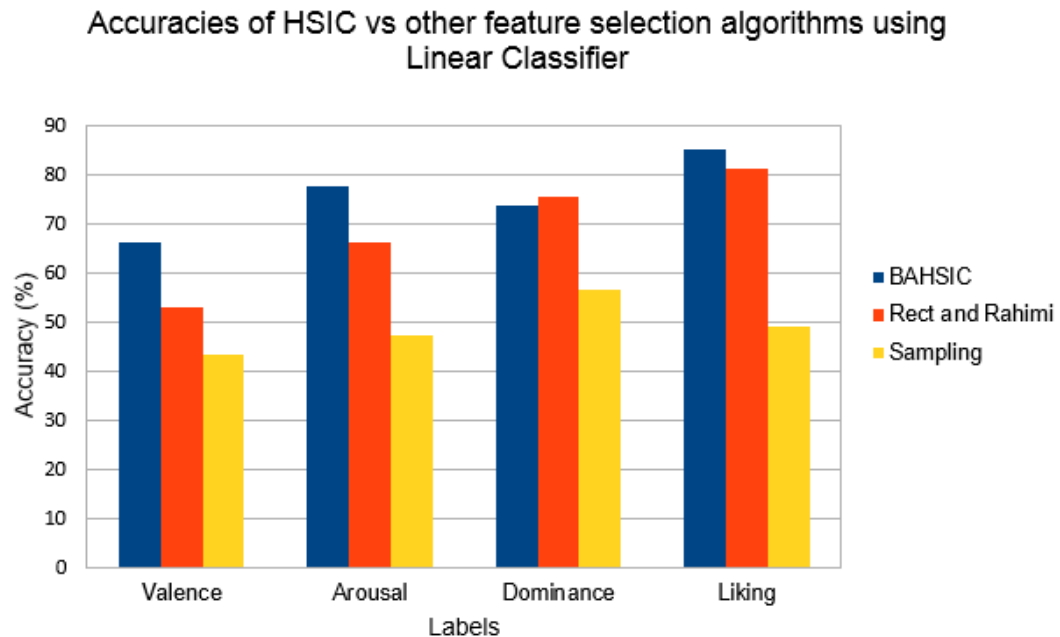
Table 2: Accuracy using Recht and Rahimi's Random Fourier Features

Benchmark	Valence	Arousal	Dominance	Liking
Linear	52.83 %	66.04 %	75.47 %	81.13%
Bayesian	84.91 %	100.00%	100.00%	88.68 %
Dtree	62.26 %	88.68 %	79.25 %	86.79 %
SVR	100.00%	100.00%	100.00%	100.00%
Logistic	45.28 %	66.04 %	75.47 %	49.06 %

Table 3: Accuracy using Down-sampling

8 Conclusion

The following chart shows the comparison of accuracies of HSIC with Sampling and Rect and Rahimi's Random Fourier Features algorithms using Linear Classifier. It is clear from the plot that HSIC performs better than Sampling and Rect and Rahimi's Random Fourier Features in Emotion Classification of EEG Signals.



9 Future Work

BAHSIC seems to perform better than Sampling and Recht and Rahimi's Random Fourier Features in Emotion Classification of EEG Signals on the subset. Validating the results on the complete dataset should give a more concrete result. With access to better computing resources, cross-validation for parameter selection can be done to improve the baselines.

10 Source Code

The following GitHub repository contains the source code of this project

<https://github.com/rishirdua/emotion-classification>

11 Open source licences

scikit-learn is a Python module for machine learning built on top of SciPy and distributed under the 3-Clause BSD license.

12 References

- [1] Song, Le, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. "Feature selection via dependence maximization." *The Journal of Machine Learning Research* 98888, no. 1 (2012): 1393-1434.
- [2] Sohaib, Ahmad Tauseef, Shahnawaz Qureshi, Johan Hagelbäck, Olle Hilborn, and Petar Jerčić. "Evaluating classifiers for emotion recognition using EEG." *In Foundations of Augmented Cognition*, pp. 492-501. Springer Berlin Heidelberg, 2013.
- [3] DEAP: A Database for Emotion Analysis using Physiological Signals", S. Koelstra, C. Muehl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, I. Patras, *IEEE Transaction on Affective Computing*, under review
- [4] *Biological Knowledge Discovery Handbook: Preprocessing, Mining and Postprocessing of Biological Data* by Mourad Elloumi, Albert Y. Zomaya, IEEE, Wiley