Preprint version. Manuscript submitted to and accepted by 2017 International Conference on Cyberworlds.

Unsupervised Feature Learning for EEG-based Emotion Recognition

Zirui Lan, Olga Sourina
Fraunhofer Institute Singapore
Nanyang Technological University
Singapore, Singapore
{LANZ0001,EOSourina}@ntu.edu.sg

Lipo Wang
School of Electrical and Electronic Engineering, Nanyang Technological University
Singapore, Singapore
ELPWang@ntu.edu.sg

Reinhold Scherer, Gernot Müller-Putz
Institute of Neural Engineering,
Graz University of Technology
Graz, Austria
{reinhold.scherer,
gernot.mueller}@tugraz.at

***Abstract*—Spectral band power features are one of the most widely used features in the studies of electroencephalogram (EEG)-based emotion recognition. The power spectral density of EEG signals is partitioned into different bands such as delta, theta, alpha and beta band etc. Though based on neuroscientific findings, the partition of frequency bands is somewhat on an ad-hoc basis, and the definition of frequency ranges of the bands of interest can vary between studies. On the other hand, it is also arguable that one definition of power bands could perform equally well on all subjects. In this paper, we propose to use autoencoder to automatically learn from each subject the salient frequency components from power spectral density estimated as periodogram by Fast Fourier Transform (FFT). We propose a network architecture especially for EEG feature extraction, one that adopts hidden unit clustering with added pooling neuron per cluster. The classification accuracy with features extracted by our proposed method is benchmarked against that with standard power features. Experimental results show that our proposed feature extraction method achieves accuracy ranging from 44% to 59% for three-emotion classification. We also see a 4-20% accuracy improvement over standard band power features.**

Keywords—emotion classification, electroencephalogram (EEG); brain-computer-interface (BCI); power spectral density; unsupervised feature extraction; autoencoder

I. Introduction

Affective brain-computer-interface (BCI) [1] envisions an emotion-aware interaction between human and machine. The goal of an affective BCI is to detect the emotion states of the user via electroencephalogram (EEG) signals and to respond to the user accordingly. Such a BCI could potentially enrich the user's experience during the interaction with a machine. EEG-based affective BCI does not rely on explicit inputs from the user, but on direct measurement of spontaneous brain activities. Thus, this modality could potentially reveal the truly-felt emotions of the user.

A closed-loop affective BCI generally consists of signal acquisition, feature extraction, neural pattern classification and feedback to the user. Feature extraction and neural pattern classification are arguably the most crucial parts in the loop. Spectral band power features have been one of the most widely used feature [2] in BCI studies and EEG-based applications. Despite their popular use, however, there lacks a consensus on the definition of frequency ranges—different studies respect different definitions. On the other hand, we argue that the most discriminative frequency components with respective to the task in question are subject-specific, that is, it is difficult to find a common definition of frequency ranges that could perform equally well on all subjects. In view of this, we propose to use autoencoder to learn from each subject the subject-specific, salient frequency components from the power spectral density of EEG signals. Building upon the trained autoencoder, we propose a network architecture especially for EEG feature extraction, one that adopts hidden neuron clustering with added pooling neuron per cluster. The classification performance using features extracted by our proposed method is benchmarked against that using band power features.

The remainder of the paper is organized as follows. Section II introduces the dataset based on which we carry out the experiment. Section III explains the methodologies. Section IV documents the experiments. Section V presents the experimental results with discussions. Section VI concludes the paper.

II. Dataset

In this study, we use a publicly available affective EEG dataset SEED contributed by Zheng *et al.* [3]. The dataset contains 15 subjects, each subject taking three recording sessions during one month at an interval of two weeks between successive sessions. In each session, each subject was presented fifteen movie clips to induce the desired emotional states: positive, neutral and negative, with five movie clips assigned to each emotion. Sixty-two-channeled (see Fig. 1) EEG signals were simultaneously recorded when the subject was exposed to the affective stimuli, at a sampling rate of 1000 Hz. The EEG signals were then down-sampled to 200 Hz and post-processed by a 0-75 bandpass filter by the authors. The same affective stimuli were used for all three sessions. The resultant dataset contains fifteen EEG trials corresponding to fifteen movie clips per subject per session. Each trial lasts for three to five minutes, depending on the length of the movie clip. The trial IDs and their corresponding emotion states are listed in Table I.

DOI: 10.1109/CW.2017.19

TABLE I. TRIAL IDS AND THEIR CORRESPONDING EMOTION STATES.

Trial IDs	Induced Emotion
1, 6, 9, 10, 14	Positive
2, 5, 8, 11, 13	Neutral
3, 4, 7, 12, 15	Negative

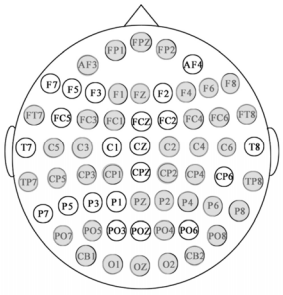

Fig. 1 The placement of 62 EEG electrodes. Shaded channels are used in this study. Other channels are rejected due to signal quality issues. (Figure adapted from [3].)

III. METHODS

Before feature extraction, we visually inspect the signal quality of all trials. Twenty-one EEG channels are rejected due to signal quality issues (e.g., loose electrode contact). The remaining 41 channels are used for further processing, including AF3, C2, C3, C4, C5, C6, CB1 (cerebellum), CB2, CP1, CP2, CP3, CP4, CP5, F1, F4, F6, F8, FC1, FC3, FC4, FC6, FP1, FP2, FPZ, FT7, FT8, FZ, O1, O2, OZ, P2, P4, P6, P8, PO4, PO5, PO7, PO8, PZ, TP7 and TP8, as shown in Fig. 1. All EEG trials except the shortest one are truncated at the end to have the same length as the shortest trial, which is 185-second. Each EEG trial is then segmented to multiple 4-second-long sections (each section equaling to 800 sampling points) without overlapping between any two successive sections. As such, each trial yields 46 sections. Features are extracted out of each section.

A. Power Feature Extraction

Band power features are one of the most widely used features in the context of EEG-based emotion recognition [2]. Though based on neuroscientific findings, the frequency band ranges of interest are somewhat defined on an ad-hoc basis and vary between studies. In our study, we follow such definition [2]: delta band (1-4 Hz), theta band (4-8 Hz), alpha band (8-12 Hz) and beta band (12-30 Hz). We does not include the gamma band (>30 Hz) in this study, as the gamma components are more artefact-prone.

Let $X \in \mathbb{R}^{s \times t}$ be one section of EEG signals, where $s = 41$ is the number of channels and $t = 800$ the number of points sampled. The power spectral density of $X(i,:)$ is estimated as periodogram by Fast Fourier Transform (FFT), where $X(i,:)$ is the ith row of X. Since one row comprises 800 points sampled at a rate of 200 Hz, the resolution of the periodogram is 0.25 Hz. The power features are computed by averaging the periodogram over the target frequency ranges defined above. The final feature vector is a concatenation of the features of the same frequency band derived from all 41 channels. The dimension of the feature vector is 41 when using delta band, theta band, alpha band or beta band alone. In addition, we also combine all power bands at feature level by concatenating the feature vectors of four bands. The feature vector is of 41×4 = 164 dimension. Each trial yields 46 samples per feature.

B. Autoencoder

An autoencoder is a neural network that is trained to produce outputs approximating to its inputs [4]. The structure of a simple feedforward, nonrecurrent autoencoder with one hidden layer is shown in Fig. 2. The objective of the autoencoder is to make $\hat{\boldsymbol{x}}$ resemble $\boldsymbol{x}$. Autoencoder can be trained using the backpropagation algorithm [4]. The training does not involve the class labels of the data and is on an unsupervised basis. However, we are not particularly interested in the output $\hat{\boldsymbol{x}}$. Instead, we are more interested in the output of hidden layer, $\boldsymbol{h}$. When the hidden layer has fewer neurons than the input layer, $\boldsymbol{h}$ is a compressed representation of $\boldsymbol{x}$ and has to capture the most salient feature of $\boldsymbol{x}$ [4] in order to be able to reproduce it at the output layer. $\boldsymbol{h}$ could then be used for further feature learning or as the feature vector of $\boldsymbol{x}$ for classification or regression.

1) Proposed Structure

In this study, we leverage autoencoder to automatically learn the salient frequency components from the periodogram instead of predefining the frequency ranges such as delta, theta, alpha and beta. The input to the autoencoder is the raw periodogram from 1 to 30 Hz with a resolution of 0.25 Hz. The dimension of the periodogram is 117-D, thus the input layer and output layer both consist of 117 neurons. The hidden layer consists of k neurons. After the autoencoder has been trained, the hidden neurons have learnt the salient frequency components over 1-30 Hz. Such information is encoded in the weight vectors of the hidden neurons. We hypothesize that hidden neurons carrying similar weights have learnt similar frequency components. We propose to cluster the hidden units into several groups by their weight vectors. A mean pooling neuron is added on top of each group to aggregate the outputs from all hidden neurons that are in the same cluster. The outputs of the mean pooling neurons are considered features learnt from the raw periodogram, which are

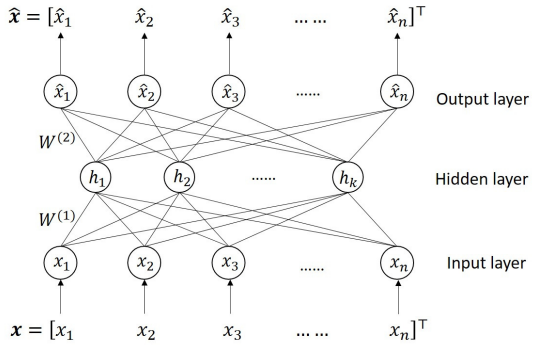

Fig. 2 Example of an autoencoder with one hidden layer. $\boldsymbol{x}$ is the input to the network, $\boldsymbol{h} = f(W^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})$ and $\hat{\boldsymbol{x}} = g(W^{(2)}\boldsymbol{h} + \boldsymbol{b}^{(2)})$, where $\boldsymbol{h}$ is the output of hidden neurons (also known as *code*), $W^{(1)}$ is the weights between hidden layer and input layer, $\boldsymbol{b}^{(1)}$ is the bias vector of hidden neurons (not drawn in the figure), $f(\cdot)$ is the activation function of hidden neurons (also known as transfer function), $W^{(2)}$ is the weights between hidden layer and output layer, $\boldsymbol{b}^{(2)}$ is the bias vector of output neurons, $g(.)$ is the activation function of output neurons. The network is trained to reproduce input $\boldsymbol{x}$ at the output layer.

DOI: 10.1109/CW.2017.19

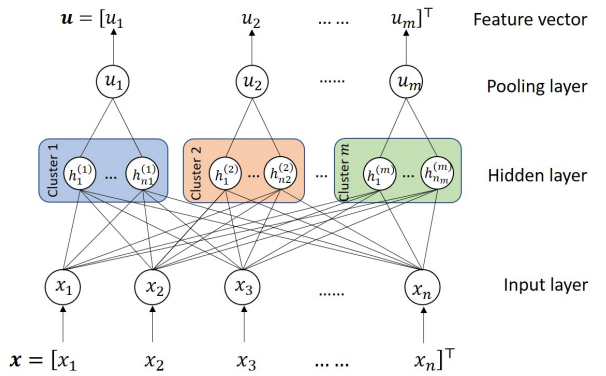

Fig. 3 Proposed network structure. After an autoencoder has been trained, k hidden neurons are clustered to m groups based on the similarity of their weights. Neurons within the same groups carry weights similar to each other, thus have learnt similar components from the input. $h_i^{(c)}$ is the output of the ith hidden neuron in cluster c. n_c is the number of neurons belonging to cluster c, $\sum_{c=1}^{m} n_c = k$. u_c is the pooling neuron added to cluster c. $u_c = \frac{1}{n_c}\sum_{i=1}^{n_c} h_i^{(c)}$. $\boldsymbol{u} = [u_1, u_2, \ldots, u_m]^\top$ is viewed as the feature extracted out of $\boldsymbol{x}$.

essentially weighted power features, but without predefinition of band ranges. The final feature vector is the concatenation of features derived from 41 channels. The dimension of the final feature vector is therefore $41m$, where m is the number of clusters of hidden neurons. The proposed network structure is illustrated in Fig. 3.

IV. Experiments

We benchmark the performance of standard power features against features that are automatically learnt by autoencoder under our proposed structure. The performance is measured by accuracy discriminating the three emotion states. Since there are 3 classes to be classified, the theoretical chance level is 33.33%.

A. Using standard power features

We evaluate the classification accuracy on a per-subject basis by five-fold cross validation. Within one session, the fifteen trials from the subject in question are partitioned into five folds as follows. Fold 1 = {trial #1,2,3}, fold 2 = {trial #4, 5, 6}, fold 3 = {trial #7, 8, 9}, fold 4 = {trial #10, 11, 12} and fold 5 = {trial #13, 14, 15}. Each fold contains one trial for each emotion. We train the classifier with four folds and test the classifier with the remaining fold. As such, the training set comprises 46×3×4 = 552 training samples and the test set consists of 46×3 = 138 test samples. The process is repeated five times until each fold has served as test set for exactly once. The per-subject classification accuracy is averaged over five runs. The overall mean accuracy is the average per-subject accuracy over fifteen subjects. In this experiment, we adopt logistic regression classifier [5] with line search strategy. The training process stops at maximal 100 iterations.

B. Using features learnt by autoencoder with the proposed structure

Firstly, we need to train the autoencoder to reconstruct the input data (periodograms). Based on the same partition scheme as used in Section IV.A, we set aside one fold as test set, and the remaining four folds are pooled together as training set. The training set comprises 46×41×3×4 = 22632 periodograms. The test set consists of 46×41×3 = 5658 periodograms. Eighty-five percent of the data randomly sampled from the training set are used as actual training data by the autoencoder, and the rest fifteen percent of the data in the training set are used as validation data to select the best weights, that is, the weight parameters that lead to the minimal reconstruction error on the validation data. In this experiment, we use one hidden layer with k = 100 hidden neurons. Input data are 117-D raw periodogram covering 1-30 Hz frequency range. Thus, the autoencoder architecture is 117 (input neurons)-100 (hidden neurons)-117 (output neurons). Linear activation function is used in all layers. The reconstruction error between input $\boldsymbol{x}$ and output $\hat{\boldsymbol{x}}$ is measured by mean squared error. The whole network is trained using backpropagation and batch gradient descent with batch size equal to 256. Training stops at maximal 50 epochs. The weight parameters that minimize the reconstruction error on validation data are retained. After the autoencoder has been trained, the output layer is removed from the network. We then employ k-means algorithm to cluster the hidden units to m groups based on the similarity of their weight vectors, m varies from 1 to 10. A mean pooling neuron is added on top of each group to aggregate the outputs, as is shown in Fig. 3. The outputs of the mean pooling neurons are viewed as features extracted out of the periodogram. The training data, validation data and test data are fed to the trained network with added pooling layers to extract features. The final feature vector is a concatenation of features from 41 channels. The classifier (same configuration as what is used in Section IV.A) is trained on training data pooled with validation data, and tested on the test data. As such, the training data and validation data together contribute 552 training samples to the classifier. The test data contribute 138 test samples. The procedures (autoencoder training, hidden unit clustering, feature extraction and classifier training and testing) are repeated five times per subject, until each fold has served as test set for exactly once. The per-subject classification accuracy is averaged over five runs. The overall mean accuracy is the average per-subject accuracy over fifteen subjects.

V. Results and Discussions

The accuracy results classifying three emotion states using different features are tabulated in TABLE II. Among the four spectral band power features, beta power performs the best. Theta and alpha powers give similar performance, both being inferior to beta and delta power. The fusion of all power features (combined power) does not lead to improved accuracy compared to beta power feature.

The results of the proposed feature extraction method are displayed at the lower half of Table II, with varying number of clusters of hidden neurons m from 1 to 10. When m = 1 and 2, the accuracy is better than that of delta, theta, alpha powers but below beta power. Starting from m = 3, the accuracy of the proposed feature exceeds standard power features. When m = 4, the feature vector dimension is the same as combined power features. The accuracy of the proposed method sees a 10.12% increase over combined power feature. There is also a tendency that the classification accuracy increases with growing number of clusters of hidden neurons. The best accuracy is attained by

DOI: 10.1109/CW.2017.19

TABLE II. OVERALL MEAN CLASSIFICATION ACCURACY (%) CLASSIFYING THREE EMOTIONS (POSITIVE, NEUTRAL AND NEGATIVE) USING DIFFERENT FEATURES.

Feature		Session 1	Session 2	Session 3	Average
Delta		43.68	40.54	41.86	42.03
Theta		43.09	40.11	39.06	40.75
Alpha		41.11	40.02	39.84	40.32
Beta		50.16	50.01	48.51	49.56
Combined power		44.03	40.55	41.89	42.16
Proposed method	$m = 1$	44.70	43.14	46.22	44.69
	$m = 2$	47.23	46.65	48.66	47.51
	$m = 3$	50.45	49.95	50.48	50.29
	$m = 4$	52.37	50.87	53.60	52.28
	$m = 5$	53.77	54.25	55.68	54.57
	$m = 6$	57.07	56.92	58.41	57.47
	$m = 7$	56.37	56.88	59.14	57.46
	$m = 8$	56.62	57.98	58.66	57.75
	$m = 9$	56.97	58.12	59.10	58.06
	$m = 10$	58.19	59.24	59.66	59.03

the proposed method when m = 10, a nearly 20% increment over theta and alpha power. However, when m exceeds 6, the improvement is only marginal. It is also worth noting that a large m value may not always be favorable, especially when the size of the training set is small. A larger m value results in large-dimensional feature vector, which requires more training data to fit the classifier. A limited training set increases the risk of overfitting when using large feature vectors.

To see what frequency components have been chosen by the autoencoder, we visualize the weights of clustered hidden units in Fig. 4. The plots show the weights of connection between input layer and hidden layer of the trained autoencoder of subject 1 in session 1 when m = 2. We average the weights within the same cluster and display the positive averaged weights for each cluster. Generally, a connection with positive weight between input neuron i and hidden neuron j suggests that hidden neuron j favors the input from neuron i, whereas negative weight implies that hidden neuron j opposes the input from neuron i. The first cluster of hidden neurons has three weight peaks at 5.5 Hz, 13.75 Hz and 24 Hz, respectively, suggesting that this cluster of hidden neurons may favour theta and beta components. The second cluster show rather evenly distributed weights over the spectrum, peaking at 8.75 Hz within the alpha band. Some delta and higher beta components are also selected by the second cluster, contrary to the first cluster.

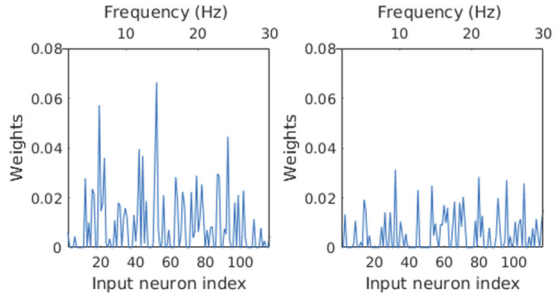

Fig. 4 Plots of averaged weights of connection between hidden neurons within the same cluster and input neurons. Left: cluster 1; right: cluster 2. Bottom horizontal axis represents the index of input neuron (117 in total). Each input neuron reveives the magnitude of periodogram at specific frequency. The frequency is noted at the top horizontal axis (1-30 Hz, corresponding to the 117-D periodogram at a resolution of 0.25 Hz). The weight indicates to what extent a specific frequency component is favoured by the hidden neuron. The left cluster has a strong preference for theta and beta components. The right cluster has preference for delta and higher theta components as compared to the left cluster.

VI. CONCLUSIONS

Spectral band power features have been one of the most widely used features in BCI studies and EEG-based applications. On the one hand, the definition of frequency range, though based on neuroscientific findings, is somewhat on an ad-hoc basis and varying between studies. On the other hand, it is arguable that one definition of band ranges could perform equally well on all subjects. In this study, we proposed to find the subject-specific salient frequency components using autoencoder. We propose a network architecture especially for power feature extraction out of raw periodograms of EEG signals. The proposed architecture consists in clustering the hidden neurons of a trained autoencoder with added pooling neuron per each cluster. The proposed method essentially extracts features similar to power features, but without predefinition of band ranges. We benchmark the proposed methods against standard power feature extraction method. Experimental results show that our proposed method yields better accuracy than standard power features when the number of hidden unit clusters $m \geq 3$. When $m = 4$, the proposed method yields feature of the same dimension as combined power features, but performs better than the latter by 10.12%. The classification accuracy can be further improved given a larger value of m, but we also see that the accuracy increment is marginal when m exceeds 6. We conclude that the proposed method, which automatically learns the salient frequency components, could potentially outperform standard band power features, whose frequency components are explicitly defined.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

REFERENCES

[1] C. Mühl, B. Allison, A. Nijholt, G. Chanel. "A survey of affective brain computer interfaces: principles, state-of-the-art, and challenges," Brain-Computer Interfaces, vol. 1, no. 2, pp. 66-84, 2014.

[2] R. Jenke, A. Peer and M. Buss, "Feature Extraction and Selection for Emotion Recognition from EEG," in IEEE Transactions on Affective Computing, vol. 5, no. 3, pp. 327-339, 2014.

[3] W. L. Zheng and B. L. Lu, "Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition with Deep Neural Networks," in IEEE Transactions on Autonomous Mental Development, vol. 7, no. 3, pp. 162-175, 2015.

[4] I. Goodfellow, Y. Bengio and A. Courville, Deep learning. Cambridge, Massachusetts: The MIT Press, 2016.

[5] F. C. Pampel. Logistic regression: A primer. Thousand Oaks, Calif: Sage Publications, Sage publications; 2000.

DOI: 10.1109/CW.2017.19