Emotion Recognition from EEG Signals

Alessio Quercia

Abstract—In this paper I show classification results for emotion recognition from EEG signals, based on the method described in the article [1]. This method consists in making classification and prediction using features extracted from decomposed EEG signals.

First of all, EEG signals taken from the DEAP dataset are decomposed into IMFs (Intrinsic Mode Functions) using the EMD (Empirical Mode Decomposition), then the first difference of time series, the first difference of phase and the normalized energy are extracted as features from the first IMF, the most informative one. Two disjoint sets are formed using the extracted features: a training set and a test set. Classification was made on the training set and the respective label set; then, prediction was made on the test set, to test the trained model on a different set without giving it the labels as input. Predicted values was confronted with the labels to check if the model prediction was right or wrong. Using cross-validation, it was possible to extimate the model accuracy for a single participant in the DEAP dataset. Finally the method accuracy was computed as the mean of the accuracies computed on the single participants.

1 Introduction

Emotions play an important role in our daily life. Indeed we use them everyday consciously or subconsciously in decision-making processes, in human interaction and for the perception of the world around us. They are also an important factor in Human-Computer Interaction (HCI), which aims at improving the communication between humans and computers, and Affective Computing, which aims at modeling emotional interactions between humans and computers by measuring users' emotional states. A person can be asked to report how he is feelin while watching a video, but the subjective self-reports alone are not enough to correctly measure his emotional state, because he may answer not exactly how he is feeling. For this reason, physiological signals can be measured to help understanding the person's emotional state. Different kinds of signals can be used for this purpose: the Galvanic Skin Response (GSR), which increases linearly with a person's level of arousal, the Electromyography (EMG), which measures the frequency of muscle tension and is correlated with negatively valenced emotions, the Heart Rate (HR), which increases with negatively valenced emotions (such as fear), the Respiration Rate (RR), which measures how deep and fast the breath is and becomes irregular with more aroused emotions (such as anger), the Electroencephalography (EEG), which has a poor space resolution, but a great time resolution, that allows to study the phase changes in response to emotional stimuli [1], [2].

There are many features extraction methods used to recognize emotions from EEG signals, including time domain techniques, frequency domain techniques and joint time-frequency analysis techniques. Also there are different types of features to extract from EEG signals, such as first and second difference, mean value and power (usually used in time domain techniques), or nonlinear features like fractal dimension (FD), sample entropy and nonstationary index.

I used the method described in the article [1]. The method consists in the following steps:

1) Sampling the EEG signals each 5 seconds.
2) EEG signal samples decomposition into Intrinsic Mode Functions (IMFs) using Empirical Mode Decomposition (EMD).
3) Features extraction from the most informative IMF of the most informative channels, forming two disjoint sets, a training set and a test set. For each IMF three features are extracted: the first difference of IMF time series, the first difference of IMF's phase and the normalized energy. These three features depict the characteristics of the IMF in time, frequency and energy domain, providing multidimensional information.
4) Classification on the training set, composed by most of the extracted features, together with their respective labels.
5) Prediction on the test set, composed only by the remaining features.
6) Recognition of the results, confronting the predicted values with the respective labels to check whether the model predicted the right value or the wrong one.

The first step is needed to obtain more samples from a single EEG signal measured on a participant watching a 60 seconds video. In the second step EMD is used to decompose the EEG signal sample into IMFs, which represent different frequency components of original signals, with band-limited characteristic. In the third step three features are extracted for each IMF. The first difference of IMF time series depicts the intensity of

Alessio Quercia, Natural Interaction and Models for Affecting Computing Courses, A/A 2017-2018, Università degli Studi di Milano, via Comelico 39/41, Milano, Italy
E-mail: alessio.quercia@studenti.unimi.it

signal change in time domain, the first difference of IMF's phase measures the change intensity in phase and the normalized energy describes the weight of current oscillation component [1].

In this paper I show the classification and prediction results obtained for emotion recognition from EEG signals using a SVM classifier based on libsvm library, with default settings and radial basis function as kernel. Model accuracy is computed for each participant in the DEAP dataset separately and finally a mean is computed to obtain the method accuracy over all the samples used.

2 State of the art

Human-Computer Interaction, Natural Interaction and Affective Computing are subject of interest of many researches. In the last years the interest on them has increased for researchers aim at improving human lifestyle quality, by improving the communication between humans and computers, since computers have an high impact on our daily life. There are several works on emotion recognition in literature, many of which use EEG signals to recognize emotional states. The articles [1], [2] on which I based my project are just two examples of these works.

3 Background

The following two subsections briefly describe what emotions and electroencephalography are and how they can be used for emotion recognition.

3.1 Emotions

An emotion is a complex psychological state that involves three distinct components: a subjective experience, a physiological response and a behavioral or expressive response [2]. Emotions are discrete and consistent responses to events with significance for the organism. These responses may include verbal, behavioral physiological and neural mechanisms. Emotions can be represented with two different perspectives:

- The *categorial perspective*, which indicates that emotions have evolved through natural selection. According to this perspective there are some basic inherited emotions, for example Plutchik proposed eight basic emotions: anger, fear, sadness, disgust, surprise, curiosity, acceptance and joy.
- The *dimensional perspective* (based on cognition), which sees the emotions mapped into the Valence, Arousal and Dominance (VAD) dimensions. Valence represents how positive is the feeling and varies from very negative to very positive; Arousal (also called Activation) indicates the level of excitement and varies from states like sleepy to excited; and Dominance corresponds to the strength of the emotion.

3.2 Electroencephalography (EEG)

EEG is a medical imaging technique that reads the scalp electrical activity generated by brain structures, i.e., it measures voltage fluctuations resulting from ionic current flows within the neurons of the brain [2].

The cortex, the largest part of the human brain, is divided into four lobes:

- the frontal lobe, responsible for the conscious thought;
- the temporal lobe, responsible for the senses of smell and sound, and the processes of complex stimuli such as faces and scenes;
- the parietal lobe, responsible for integrating sensory information from various senses and for the manipulation of objects;
- the occipital lobe, responsible for the sense of sight.

The signals observed with the EEG can be divided into specific ranges:

- the delta waves (1-4 Hz), associated with the unconscious mind;
- the theta waves (4-7 Hz), associated with the subconscious mind;
- the alpha waves (8-13 Hz), associated to a relaxed mental state;
- the beta waves (13-30 Hz), related to an active state of mind;
- the gamma waves (¿30 Hz), associated with an hyper brain activity.

There are two main areas of the brain correlated with emotional activity: the *amygdala* (located close to the *hippocampus*, in the frontal portion of the temporal lobe) and the pre-frontal cortex (part of the frontal lobe). The frontal and parietal lobes are the most informative about the emotional states, while the alpha, gamma and beta appear to be the most discriminative [2].

4 Theoretical model

4.1 Dataset

The DEAP dataset consists of two parts:

1) The ratings from an online self-assessment where 120 one-minute extracts of music videos were each rated by 14-16 volunteers based on arousal, valence and dominance.
2) The participant ratings, physiological recordings and face video of an experiment where 32 volunteers watched a subset of 40 of the above music videos. EEG and physiological signals were recorded and each participant also rated the videos as above. For 22 participants frontal face video was also recorded.

I used the preprocessed data contained in the second part, which consists of 32 files, corresponding to the 32 experiment's participants. These files contain a downsampled (to 128Hz), preprocessed and segmented version of the original data. Each participant file contains two arrays, as shown in the Table 1.

Array name	Array shape	Array contents
data	40 x 40 x 8064	video/trial x channel x data
labels	40 x 4	video/trial x label (valence, arousal, dominance, liking)

TABLE 1: DEAP dataset.

4.2 Sampling from the dataset

The preprocessed EEG signals in the DEAP dataset have been sampled each 5 seconds to have bigger training and test sets. Indeed splitting the signals each 5 seconds results in having 12 samples per video, each containing 40 different signals, corresponding to 40 different channels. Repeating this for all the 40 videos proposed to each participant in the dataset, it was possible to have 480 samples per participant. The training set was formed using 468 samples, corresponding to 39 videos, and the respective labels (repeated for the 12 split samples); while the test set was formed using 12 samples, corresponding to the remaining video.

4.3 Empirical Mode Decomposition (EMD)

Each EEG signal sample (from here on just 'sample') is decomposed into a set of Intrinsic Mode Functions (IMFs) by Empirical Mode Decomposition (EMD). Each IMF represents different frequency components of original signals.

For input signal $x(t)$, the EMD process is:

1) Set $h(t) = x(t)$ and $h_{old}(t) = h(t)$.
2) Get local maximum and minimum of $h_{old}(t)$.
3) Interpolate the local maximum and minimum with cubic spline function and get upper envelope $e_{max}(t)$ and and lower envelope $e_{min}(t)$.
4) Calculate the mean value of the upper and lower envelope as

$$m(t) = \frac{e_{min}(t) + e_{max}(t)}{2}. \quad (1)$$

5) Subtract $h_{old}(t)$ with $m(t)$:

$$h_{new}(t) = h_{old} - m(t). \quad (2)$$

If $h_{new}(t)$ satisfies the following two conditions:

a) during the whole data set, the number of extreme points and the number of zero crossing must be either equal or differ at most by one;
b) at each point, the mean value calculated from the upper and lower envelope must be zero; then the first IMF component imf_1 is gotten; otherwise, set $h_{old}(t) = h_{new}(t)$ and go to step (2), repeating steps (2)-(5) until $h_{new}(t)$ satisfies the two conditions of IMF. Finally imf_1 is gotten as

$$imf_1 = h_{new}(t). \quad (3)$$

6) If imf_n is gotten, set h_{old} as

$$h_{old}(t) = h_{old}(t) - imf_n. \quad (4)$$

Then go to step (2) and repeat steps (2)-(5) to get imf_{n+1}.

The signal $x(t)$ can be expressed as

$$x(t) = \sum_{n=1}^{L} imf_n + r, \quad (5)$$

a linear combination of IMF components and the residual part r [1].

4.4 Features extraction

For each sample, three features have been extracted from the most informative IMF (the first) of the most (8 channels) informative electrodes (Fp1, Fp2, F7, F8, T7, T8, P7, and P8). Indeed 24 features for each sample were collected. The three features extracted for the first imf of each of the above mentioned channels are the following [1]

1) *First difference of IMF time series*, which depicts the intensity of signal change in time domain. For an IMF component with N points, IMF$\{imf_1, imf_2, ..., imf_N\}$, it can be computed as follows:

$$D_t = \frac{1}{N-1} \sum_{n=1}^{N-1} |imf(n+1) - imf(n)|. \quad (6)$$

2) *First difference of IMF's phase*, which reveals the change intensity of phase and represents the physical meaning of instantaneous frequency. For an N-point IMF, IMF$\{imf_1, imf_2, ..., imf_N\}$, Hilbert transform is applied to it, obtainig the analytic signal $z(n)$:

$$z(n) = x(n) + jy(n). \quad (7)$$

An equivalent way of representing the analytic signal is:

$$z(n) = A(n) + e^{j\varphi(n)}, \quad (8)$$

where $A(n) = \sqrt{x(n)^2 - y(n)^2}$ is the amplitude of $z(n)$ and $\varphi(n) = arctan(y(n)/x(n)$ is the instantaneous phase.
Finally, the first difference of IMF's phase can be computed as follows:

$$D_p = \frac{1}{N-1} \sum_{n=1}^{N-1} |\varphi(n+1) - \varphi(n)| \quad (9)$$

3) *Normalized energy of IMF*, which describes the weight of current oscillation component. For an N-point IMF, IMF$\{imf_1, imf_2, ..., imf_N\}$, it can be computed as follows:

$$E_{norm} = \frac{\sum_{n=1}^{N} imf^2(n)}{\sum_{n=1}^{N} s^2(n)}, \quad (10)$$

where $s(n)$ is the original EEG signal.

.

4.5 Classification and prediction

Once extracted the features from the EEG signals and created a training set and a test set, I fed the first one into a SVM classifier, using standard parameters and the radial basis function as kernel, and trained it. Then I used the trained model to predict emotional states using the test set as input.

Given a training set of instance-label pairs (x_i, y_i), $i = 1, ..., l$ where $x_i \in \mathcal{R}^n$ and $y_i \in \{-1, 1\}^l$, a SVM requires the solution of the following optimization problem:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + \mathcal{C} \sum_{i=1}^{l} \xi_i \tag{11}$$

subject to

$$y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \tag{12}$$

$$\xi_i \geq 0. \tag{13}$$

where $\mathcal{C} > 0$ is the penalty parameter of the error term. The training vectors x_i are mapped into a higher (maybe infinite) dimentional space by the function ϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher dimentional space [3].

A Support Vector Machine (SVM) is a supervised learning model with an associated learning algorithm that analyze data used for classification and regression analysis. Given a set of training examples, each one marked as belonging to one or the other of two categories, a SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. 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 like outliers detection.

It often happens that the sets to discriminate are not linearly separable. SVMs can efficiently perform a non-linear classification using the *kernel trick*, implicitly mapping their inputs from finite-dimensional space into higher dimentional features spaces. The kernel trick consists of observing that many machine learning algorithms can be written exclusively in terms of dot products between examples. For example:

$$w^T x + b = b + \sum_{i=1}^{m} \alpha_i x^T x^{(i)} \tag{14}$$

where $x^{(i)}$ is a training example and α is a vector of coefficients. Rewriting the learning algorithm this way allows us to replace x by the output of a given feature function $\phi(x)$ and the dot product with a function $k(x, x^{(i)}) = \phi(x) \cdot \phi(x^{(i)})$, called *kernel function*.

After replacing dot products with kernel evaluations, we can make classification using the function

$$f\big(x^{(i)}\big) = sgn\left(\left[\sum_{i=1}^{m} \alpha_i y_i k(x_i, x^{(i)})\right] - b\right) \tag{15}$$

and prediction using the function

$$f(x) = b + \sum_{i=1}^{m} \alpha_i k(x_i, x^{(i)}). \tag{16}$$

which is nonlinear with respect to x, but the relationship between $\phi(x)$ and $f(x)$ and the one between α and $f(x)$ are both linear. Using the kernel function means preprocessing the data by applying $\phi(x)$ to all inputs and then learning a linear model in the new transformed space.

The most commonly used kernel is the *Gaussian kernel*, also called *Radial Basis Function* (RBF) kernel, because its value decreases along lines in v space radiating outward from u:

$$k(u, v) = \mathcal{N}(u - v; 0, \sigma^2 I) \tag{17}$$

where $\mathcal{N}(x; \mu, \Sigma)$ is the standard normal density. The Gaussian kernel corresponds to a dot products in an infinite-dimensional space. We can think it as performing a kind of *template matching*: a training example x associated with training label y becomes a template for class y. When a test point x' is near x according to Euclidean distance, the Gaussian kernel has a large response, indicating that x' is very similar to x template. The model then puts a large weight on the associated training label y [4]. The radial basis kernel can be computed as follows:

$$k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0, \tag{18}$$

5 Simulations and Experiments

According the method described in the article [1], I used the *cross-validation* to compute the prediction accuracy for each participant, extracting features from 39 videos to form the training set, and using the remaining video to create the test set. Looping over the videos and changing each time the video to leave out for the test set, it was possible to compute 40 classifications and 40 predictions for each participant. This allowed me to compute the prediction accuracy for each participant. Then I computed the general method accuracy as the mean of the prediction accuracies.

5.1 System architecture

I ran the simulations on my laptop, characterized by an Intel Core i7-6700HQ CPU, a Nvidia Geforce GTX 950M GPU and 8 GB RAM. The implemented algorithm for emotion recognition uses the CPU only.

5.2 Implementation details

The implementation is based on the general method scheme represented in the Figure 1.

The implemented algorithm also takes in consideration the *cross-validation* as shown in the pseudo-code in the Algorithm 1.

I implemented the algorithm in python, importing *EMD* library from *PyEMD* for the Empiric Mode Decomposition, and *svm* from *sklearn* for the SVM classifier. I used the SVC classifier, based on libsvm, with standard parameters and radial basis function as kernel.

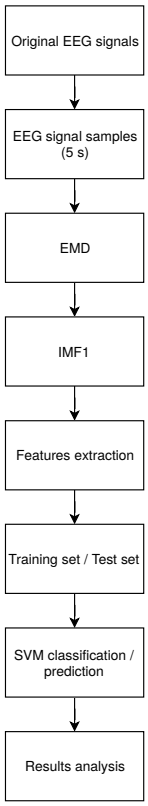

Fig. 1: Method scheme.

Algorithm 1 Emotion recognition pseudo code

```
for each participant in dataset do
  data = dataset[participant]['data']
  labels = dataset[participant]['labels']
  for each video in data do
    features_per_video = extract_features(video, data)
  leave_out = len(data) - 1
  while leave_out >= 0 do
    for each video in data do
      if video == leave_out then
        continue
      X.append(features_per_video(video))
    LABELS = ['Valence', 'Arousal']
    for each lab in LABELS do
      for each sample do
        Y = labels[video][lab]
      training_set = [X, Y]
      SVM.fit(training_set)
      test_set.append(features_per_video(leave_out))
      SVM.predict(test_set)
```

Participant	Valence Accuracy	Arousal Accuracy
1	42.5%	47.5%
2	62.5%	57.5%
3	57.5%	80%
4	62.5%	57.5%
5	65%	37.5%
6	70%	37.5%
7	55%	50%
8	47.5%	55%
9	67.5%	65%
10	75%	60%
11	25%	75%
12	62.5%	82.5%
13	72.5%	85%
14	70%	62.5%
15	65%	62.5%
16	70%	65%
17	47.5%	60%
18	65%	57.5%
19	62.5%	67.5%
20	67.5%	77.5%
21	52.5%	80%
22	50%	62.5%
23	75%	70%
24	45%	82.5%
25	45%	70%
26	65%	32.5%
27	72.5%	65%
28	70%	52.5%
29	50%	67.5%
30	62.5%	45%
31	65%	60%
32	40%	57.5%
Total	59.5%	62%

TABLE 2: Valence and arousal accuracies for each participant. In the last row the general method accuracies for valence and arousal are indicated (computed as the mean of all the the accuracies of the same emotion).

6 Results

The emotion recognition accuracy, considering the valence and the arousal, for each participant is showed in the Table 2. Valence accuracy for each participant is represented in the graph in the Figure 2, while arousal accuracy for each participant is represented in the graph in the Figure 3. Results for emotion recognition using the method previously described indicate that a mean model trained with 39 videos will recognize the correct emotion felt (predicting the correct label) with about 60% accuracy. Indeed, as shown in the Figure 4, the mean valence prediction accuracy is 59.5%, while the mean arousal prediction accuracy is 62%, considering all the participants in the DEAP dataset.

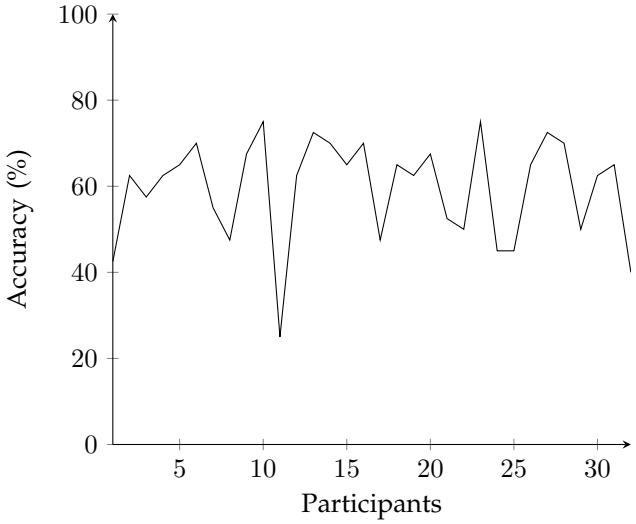

Fig. 2: Valence accuracy for each participant.

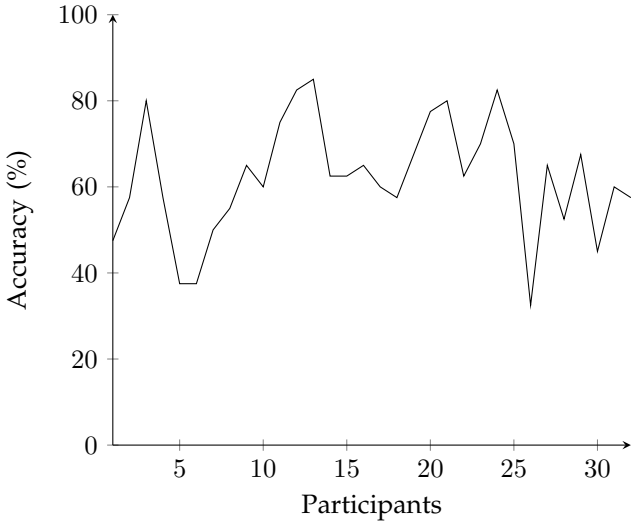

Fig. 3: Arousal accuracy for each participant.

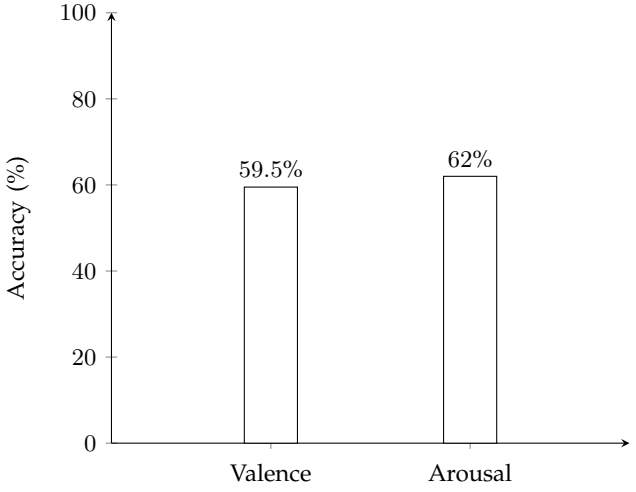

Fig. 4: Method accuracy.

7 Conclusions

The method proposed in the article [1] has proven to be a good method for emotion recognition, even if the results I obtained are slightly different from the one obtained by the method authors. Indeed it recognizes the correct emotion with an accuracy of about 60%, even if it can be boosted up to have an accuracy of about 70%.

References

[1] N. Zhuang, Y. Zeng, L. Tong, C. Zhang, H. Zhang, and B. Yan, "Emotion recognition from eeg signals using multidimensional information in emd domain," *BioMed research international*, vol. 2017, 2017.

[2] S. M. Alarcao and M. J. Fonseca, "Emotions recognition using eeg signals: A survey," *IEEE Transactions on Affective Computing*, 2017.

[3] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.