b
UNIVERSITÄT
BERN

UNIVERSITÄT
BERN

AI for Medical Time Series - Spring 2022

Group Project – Classification of EEG time series of responses to neutral vs unpleasant images

Jiahui Yu / Shunyu Wu

1 June 2022

Table of Contents

UNIVERSITÄT BERN

Introduction

Classification Task

- This dataset consists of 21 subjects, 64 channels, 384 time points. There are 6 classes representing EEG responses to images with different familiarity and pleasantness *(familiar and neutral (FN), familiar and pleasant (FP), familiar and unpleasant (FU), novel and neutral (NN), novel and pleasant (NP), novel and unpleasant (NP) pictures)*.

- The goal of the project is to train a classifier that can classify Familiar (FN, FU and FP) and Novel (NN, NP and NU).

Classification 1

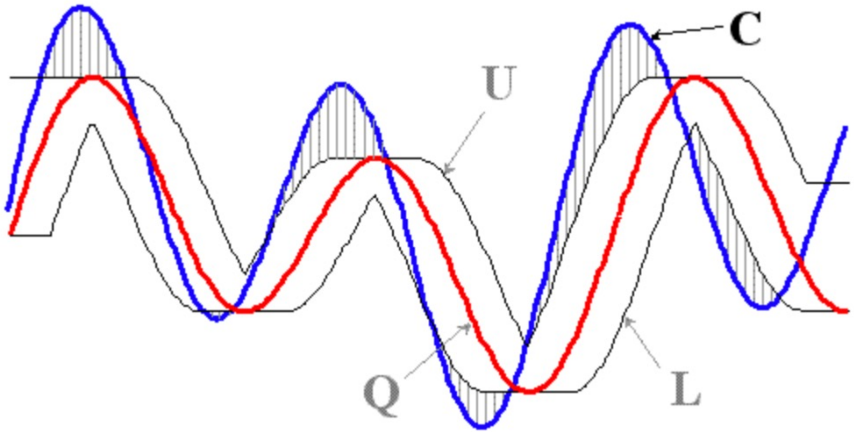
Dynamic Time Warping

- Dynamic time warping finds the optimal non-linear alignment between two time series.
- $D(n, m) = \min(D(n-1, m-1), D(n-1, m), D(n, m-1)) + c(x_n, y_m)$
  - DTW is quadratic in the length of the time series used, time complexity of O(nm)

- Speed up
  - Enforce a locality constraint window size w, cause it is unlikely for X_i and Y_j to be matched if i and j are too far apart.
  - Use the LB Keogh lower bound of dynamic time warping. It is still the fastest known technique for indexing DTW.

What is LB_Keogh?

$$LB\_Keogh(Q,C)=\sum_{i=1}^{n}\begin{cases}(q_i-U_i)^2 & \text{if } q_i>U_i\\ (q_i-L_i)^2 & \text{if } q_i<L_i\\ 0 & \text{otherwise}\end{cases}$$



To the left is a visual intuition of LB_Keogh, a protective envelope is built around the red time series, the Euclidean distance between the blue time series and the closest part of the protective envelope is a (tight) lower bound to the DTW. For indexing under uniform scaling or processing of streaming time series etc, the definition of the envelope differs, but the LB_Keogh definition is unchanged.

Classification 1

K-NN algorithm

- 1-NN algorithm with dynamic time warping Euclidean distance
- LB_Keogh(X,Y) ≤ DTW(X,Y), and computing *LB Keogh* is much less expensive than performing DTW
- In order to eliminate many unnecessary dynamic time warping computations, use LB Keogh to get rid of those cannot possibly be more similar that the current most similar time series.
- Choose window size of 4.

Classification 1

Data Preprocess

- Even though the code is sped up, it still take much longer than I expected. So I just average the channels of one subject, which is bad practice, to get an experimental result(over 1h running time for the first two subjects).

- To classify familiar vs novel, new event label 1 to familiar and 0 to novel are reassigned manually.

- Subject 1 has 983 trails and subject 2 has 268 trails, so the final data matrix size is (983+269)*(384+1). The last column is class label.

- The data set is then split into train and test using train test split from sklearn, in which 33% will be test set and 67% will be train set.

Classification 1

Result

TABLE II: classification report

	precision	recall	f1-score	support
novel: 0	0.24	0.24	0.24	91
familiar: 1	0.79	0.78	0.78	322
accuracy			0.66	413
macro avg	0.51	0.51	0.51	413
weighted avg	0.66	0.66	0.66	413

As shown left using just 2 subjects data and average channels is clearly not enough for accuracy, twhich is only 66%.

But it is enough to present that DTW and KNN did their jobs, so the idea works. Just need better CPU or GPUs to compute the whole data set.

Classification 2

Data Preprocess

- Dataset: 1-P-cleaned.fif, 2-P-cleaned.fif

Familiar (FN, FU, FP) VS Novel (NN, FP, NU)

```
[ ] label_mapping = {'FN': 0, 'FP': 0, 'FU': 0, 'NN': 1, 'NP': 1, 'NU': 1}
    dataF['condition'] = dataF['condition'].replace(label_mapping)
    dataF['condition'].value_counts()

    0    377472
    1    102912
    Name: condition, dtype: int64
```

Classification 2

Data Preprocess

- Principle Component Analysis (PCA) : 20 main components

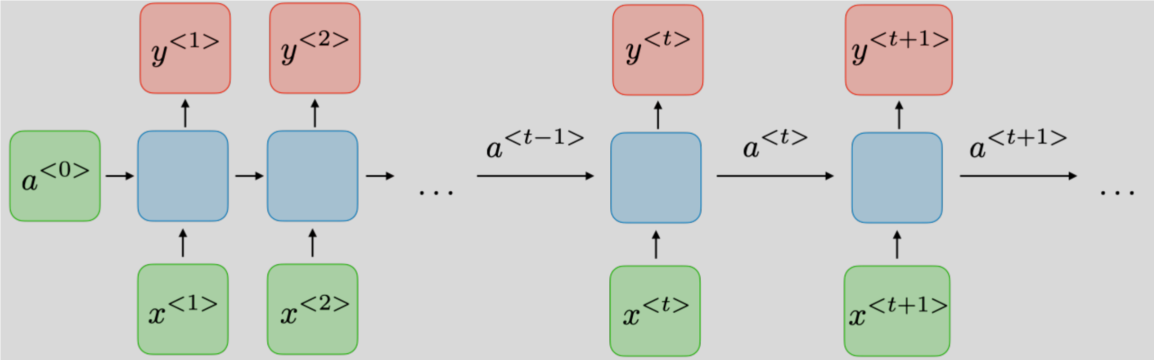
  1) Normalize all the data points
  2) Calculate the covariance matrix X of data points
  3) Calculate eigenvectors and corresponding eigenvalues
  4) Sort the eigenvectors according to their eigenvalues in decreasing order
  5) choose the first k eigenvectors, and that will be the new k dimensions
  6) Transform the original n-dimensional data points into k dimensions

- Train – Test – Validation Split
  - 60% Trainset
    - 80% for train, 20% for validation
  - 40% Test-set

Classification 2

Method

UNIVERSITÄT BERN



Input:
Data points of 20 channels in objects 1 and 2
Output:
0 (Familiar)
1 (Novel)

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

$$\Gamma = \sigma(W_x^{<t>} + U_{a^{<t-1>}} + b)$$

- update gate Γ*u* to measure how mush past should maintain
- relevance gate Γ*r* to measure how much past should drop.

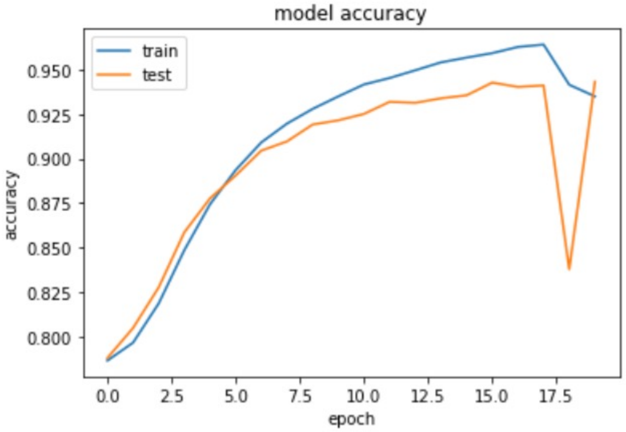
Classification 2

Method

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 20, 1)]           0

 gru (GRU)                   (None, 20, 256)           198144

 flatten (Flatten)           (None, 5120)              0

 dense (Dense)               (None, 2)                 10242

=================================================================
Total params: 208,386
Trainable params: 208,386
Non-trainable params: 0
_________________________________________________________________
```

Classification 2

Result



	precision	recall	f1-score	support
novel: 0	0.95	0.98	0.96	150907
familiar: 1	0.91	0.81	0.86	41247
accuracy			0.94	192154
macro avg	0.93	0.90	0.91	192154
weighted avg	0.94	0.94	0.94	192154

Test Accuracy : 94.225%

Open Problems

1. Time cost and Computation cost
2. Feature extraction

Reference

- 1. https://www.cs.ucr.edu/~eamonn/LB_Keogh.htm
- 2. Course slide: AI for Medical Time Series Data Lecture 6: Unsupervised learning for time-series data

Thank you

For your attention

UNIVERSITÄT
BERN

AI for Medical Time Series – Spring 2022

01.06.2022 Jiahui Yu / Shunyu Wu