P300 Speller with patient with ALS

Progetto MOBD 2018/19

Conidi Andrea, Falvo Simone

Introduzione

Lo scopo del progetto è quello di predire correttamente i caratteri corrispondenti a sequenze di stimolazioni provenienti da un'interfaccia BCI.

Il lavoro svolto è stato quello di utilizzare tecniche di classificazione binaria per distinguere le istanze *target* da quelle *non target*. In particolare è stato utilizzato un classificatore di tipo SVM lineare con l'ausilio di tecniche di data augmentation e feature selection, inoltre è stata modificata la funzione di decisione in modo che, in corrispondenza di ogni iterazione di stimolazioni, vengano selezionate come target una sola riga ed una sola colonna.

Il classificatore è stato addestrato su una porzione del dataset fornito, l'altra porzione è stata utilizzata come test ed i risultati finali hanno mostrato un'accuratezza del 100% sul test set e in cross-validazione.

Data Understanding

Per prima cosa si è tentato di avere una panoramica generale sulle classi del dataset, pertanto sono stati graficati i segnali registrati per elettrodo, distinguendo le forme d'onda relative alle istanze *target* e *non target* ed effettuando una media su di esse in modo da ridurre il più possibile il rumore.

La figura 1 mostra chiaramente la componente P300 (in rosso) che si distingue dalla stimolazione media legata ai caratteri non target che non mostra un particolare andamento.

Data Splitting

Per addestrare e testare il modello, il dataset è stato suddiviso in training set e test set con una proporzione 70-30.

Prima di effettuare lo split, è stata applicata una permutazione a blocchi di 120 righe, in modo da mantenere la sequenzialità delle iterazioni per ogni carattere, che viene poi sfruttata per determinare gli indici di riga e colonna corrispondenti al carattere target.

Data Augmentation

Per aumentare l'accuratezza del modello, sono stati introdotti dati fittizi generati a partire da un sottoinsieme del training set.

Inizialmente si è provato a generare dati aggiungendo semplicemente del rumore bianco oppure effettuando piccole traslazioni temporali dei valori di tensione (in modo da rendere più "robusto" il modello a eventuali errori o ritardi di misurazione), ma ciò non ha prodotto buoni risultati in cross-validazione. La tecnica risultata efficace è stata quella di generare nuovi caratteri aventi come valori di misurazione la media delle istanze appartenenti alla stessa classe di due caratteri diversi.

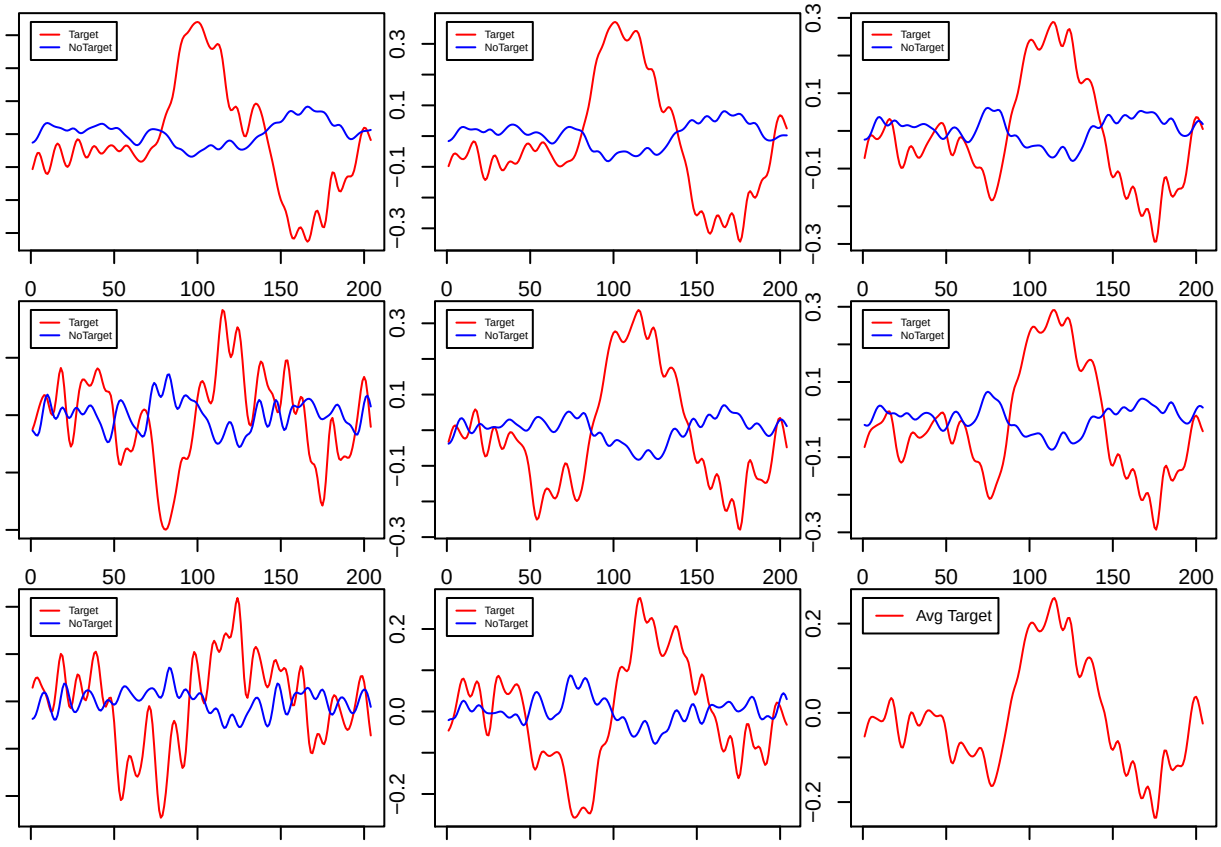

Figure 1: Panoramica stimolazioni target (linea rossa) e non target (linea blu) per gli otto elettrodi, l'ultimo grafico (in basso a destra) rappresenta la media degli stimoli target su tutti gli elettrodi

Sperimentalmente si ottengono risultati migliori generando un numero di caratteri pari al 30% di quelli presenti nel dataset, ed ognuno di essi viene generato "mediando" il numero minimo di caratteri, perché all'aumentare di tale quantità si ottengono caratteri con meno rumore che differiscono troppo dal dataset originale.

Feature Selection

In questa fase si è provato ad estrarre dai dati a disposizione nuove feature che potessero essere rilevanti ai fini della classificazione, a tale scopo è stato utile il lavoro di Abootalebi, Moradi, and Khalilzadeh (2009), tra le feature proposte si è deciso di provare a utilizzare:

- Area Positiva: somma integrale dell'area positiva sottesa dalla curva
- Area Negativa: somma integrale dell'area negativa sottesa dalla curva
- Area assoluta: somma dei moduli delle due aree precedenti
- Crossing Zero: numero di volte in cui il segnale passa per zero
- Potenza del segnale: cacolata come potenza $X^T X$, dove X è il vettore delle misurazioni
- Valore di Picco Picco: differenza tra valore massimo e valore minimo di tensione
- Time Window: intervallo di tempo tra il picco positivo e negativo

Alcune delle feature sono invece state implementate analizzando i dati nella fase di data understanding:

- Rising Time: misura dell'intravallo temporale in cui il segnale è crescente
- Correlazione P300: correlazione del segnale rispetto alla P300 ideale (estratta tramite dalle istanze target del training set)
- C_{bin}: matrice binaria calcolata a partire dal file "C.txt", dove l'elemento $C_{bin}[i, j] = 1$ se $C[i] = j$

La scelta delle features è stata fatta valutando le varie combinazioni in cross validation. Il set migliore è stato quello composto da: Area Assoluta, P300 e Crossing Zero.

Normalizzazione

A seguito dell'introduzione di nuove feature i dati vengono normalizzati applicando la seguente trasformazione su ogni colonna dei dataset:

$$Z_{train} = \frac{X_{train} - \overline{X}_{train}}{\sigma_{train}} \qquad Z_{test} = \frac{X_{test} - \overline{X}_{train}}{\sigma_{train}}$$

Dove $\overline{X}$ e σ sono rispettivamente media campionaria e deviazione standard dei dati delle istanze relativi ad una feature.

I dati di test vengono scalati utilizzando media campionaria e deviazione standard dei dati di training, essendo quest'ultimi un campione più numeroso della stessa distribuzione.

Training

Nella fase iniziale del progetto sono stati provati vari classificatori proposti da Manyakov et al. (2011), ma i risultati migliori in cross-validazione si sono ottenuti con una SVM lineare. In particolare si è scelto di impiegare l'algoritmo *liblinear* con i valori dei parametri type e cost rispettivamente pari a 7 e 0.01.

Funzione di decisione

Funzione basata sul massimo invece che sul segno. Partendo dai decision values ottenuti dalla predict di liblinear si seleziona un blocco di 120 righe corrispondente a un carattere e ogni 12 righe vengono estratti gli indici riga e colonna contenententi il valore massimo. Il risultato per ogni carattere sarà una matrice di indici riga/colonna target, con un numero di righe pari al numero di iterazioni. A tale matrice viene applicata la moda per colonne ottenendo la combinazione più frequente. L'assegnazione di pesi non uniformi alle iterazioni non produce risultati migliori, in particolare, dando più "importanza" alle prime o alle ultime o alle iterazioni centrali si ottengono valori minori o uguali di accuratezza. Infine si genera il nuovo blocco di predizioni impostando come target le istanze corrispondenti agli indici selezionati.

Cross-Validazione

Tutti i parametri del progetto sono stati calibrati sulla base dei risultati ottenuti in fase di cross-validazione.

In questa fase è stata applicata la tecnica *K-fold* con un valore di k pari a 10.

Di seguito i risultati che si ottengono:

```
print(cv_results)
```

```
##       Run Accuracy
##  [1,]   1        1
##  [2,]   2        1
##  [3,]   3        1
##  [4,]   4        1
##  [5,]   5        1
##  [6,]   6        1
##  [7,]   7        1
##  [8,]   8        1
##  [9,]   9        1
## [10,]  10        1
```

Risultati e Conclusioni

Infine il modello viene valutato sul test set, fornendo un'indicazione della percentuale di caratteri correttamenti classificati.

I risultati mostrano sul test un'accuretezza pari al 100%, tale risultato è stato ottenuto con un seme iniziale pari a 123.

```
accuracy <- test_accuracy(model, scaled_data$test)
printf("Caratteri predetti correttamente: %.2f%%", accuracy * 100)
```

```
## Caratteri predetti correttamente: 100.00%
```

How-To Test

Per e testare il modello con un nuovo test set bisogna accedere al file "main_test.R" e sostituire i nomi dei file con quelli del test. In questo caso il modello viene addestrato su tutto il training set a disposizione. L'esecuzione può richiedere qualche minuto poiché la generazione delle feature è un calcolo molto oneroso.

```
#dfx_test <- read.table("X_test.txt", header = FALSE)
#dfc_test <- read.table("C_test.txt", header = FALSE)
#dfy_test <- read.table("Y_test.txt", header = FALSE)
```

Riferimenti Bibliografici

Abootalebi, Vahid, Mohammad Hassan Moradi, and Mohammad Ali Khalilzadeh. 2009. "A New Approach for Eeg Feature Extraction in P300-Based Lie Detection." *Computer Methods and Programs in Biomedicine* 94 (1): 48–57. doi:https://doi.org/10.1016/j.cmpb.2008.10.001.

Manyakov Nikolay V., Combaz Adrien, Chumerin Nikolay, and Van Hulle Marc M. 2011. "Comparison of Classification Methods for P300 Brain-Computer Interface on Disabled Subjects." *Computational Intelligence and Neuroscience* 2011. doi:https://doi.org/10.1155/2011/519868.