Pattern Recognition in EEG

Pieter-Jan Kindermans,
UGent,
Department of Electronics and Information Systems (ELIS)
Who is familiar with machine learning?
Who is familiar with MATLAB?
Who knows how to program?
We are

Thibault Verhoeven, Pieter-Jan Kindermans

- Faculty of engineering and architecture

- Department of Electronics and Information Systems (ELIS)

- Reservoir Lab (a Machine learning group)

- PhD students

- Work on/related to Brain-Computer Interfaces
To illustrate basic machine learning principles
Outline

- Event-Related Potential classification (the task)

- Machine learning methods (the basic tools)

- Unsupervised classification in BCI (advanced tools)

- The hands on session (the work)

- Your own data?
Event-Related Potential classification (the task)

focus on ERPs in Brain-Computer Interfaces
Application: Brain-Computer Interfaces

brain-signals → decoder → application
Event-Related Potentials (Oddball paradigm)
ERP based BCI
General principle behind ERP based BCI
General principle behind ERP based BCI

Stimulus 1 2

EEG/Response
General principle behind ERP based BCI

Stimulus 1 2 3

EEG/Response
General principle behind ERP based BCI

Stimulus 1 2 3

EEG/Response

1 iteration
General principle behind ERP based BCI

1 iteration
General principle behind ERP based BCI

1 iteration
General principle behind ERP based BCI

Stimulus 1 2 3 3 1 2 3 1 2

EEG/Response

1 iteration

1 trial
General principle behind ERP based BCI

Stimulus

EEG/Response

Attended stimulus?
ERP variations

All these variations exhibit the same *stimulus/iteration structure*

- Visual speller
- Auditory (e.g. Amuse, PASS2D)
- Tactile
- ...
Example: auditory ERPs

A - supervised blocks

ssAUC

B - unsupervised blocks

Cz (thick)
F5 (thin)
Many differences between subjects
Unfortunately, the raw data looks like this
ERP Speller: The default approach

1. Record training data *(quite boring)*

2. Machine learning magic *(supervised)*

3. Use the BCI
Questions?
We will build a decoder to discriminate between target and non-target ERP responses

It is already implemented.
If you get bored, you can extend the implementation such that it predicts the symbols as well.
Machine learning methods (the basic tools)
Machine learning rules

- **Do not** optimise the model on the data used for evaluation
Machine learning rules

- **Do not** optimise the model on the data used for evaluation

- Keep the model as simple as possible
Machine learning rules

- **Do not** optimise the model on the data used for evaluation

- Keep the model as simple as possible

- Use a proper cost function
Machine learning rules

- **Do not** optimise the model on the data used for evaluation

- Keep the model as simple as possible

- Use a proper cost function

- **Do not** directly interpret the classifier weights
Linear Discriminant Analysis

Pictures from Pattern Recognition and Machine Learning (C. Bishop)
Linear Discriminant Analysis

Pictures from Pattern Recognition and Machine Learning (C. Bishop)
Linear Discriminant Analysis

\[ p(x|C_1) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu_1)^T \Sigma^{-1} (x - \mu_1)\right) \]

\[ p(x|C_2) = \frac{1}{(2\pi)^{D/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (x - \mu_2)^T \Sigma^{-1} (x - \mu_2)\right) \]

\[ p(C_1) = \pi_{C_1}, \quad 0 \leq \pi_{C_1} \leq 1 \]

\[ p(C_2) = 1 - \pi_{C_1} \]
Linear Discriminant Analysis

\[ wx + w_0 > 0 \]

\[ w = \Sigma^{-1}(\mu_1 - \mu_2) \]

\[ w_0 = -\frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 + \log \frac{p(C_1)}{p(C_2)} \]
Linear Discriminant Analysis

\[
wx + w_0 > 0
\]

\[
w = \Sigma^{-1}(\mu_1 - \mu_2)
\]

\[
w_0 = -\frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 + \log \frac{p(C_1)}{p(C_2)}
\]
Linear Discriminant Analysis

\[ wx + w_0 > 0 \]

\[ w = \Sigma^{-1} (\mu_1 - \mu_2) \]

\[ w_0 = -\frac{1}{2} \mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2} \mu_2^T \Sigma^{-1} \mu_2 + \log \frac{p(C_1)}{p(C_2)} \]
Overfitting and regularisation
Regularisation for LDA

Estimating covariance matrices is difficult (especially for high dimensions)
Shrinkage regularisation

\[ \hat{\Sigma} = \Sigma + \lambda I \]

Effect: the weight vector becomes equal to the difference between the class means:

\[ \mathbf{w} = \hat{\Sigma}^{-1}(\mu_1 - \mu_2) \]
Training and testing

data

fold 1
fold 2
fold 3
fold 4
fold 5

test
Training and testing

data

train validation test

fold 1

fold 2

fold 3

fold 4

fold 5
Crossvalidation

data

train  test

fold 1
fold 2
fold 3
fold 4
fold 5
Nested cross-validation

For all the inner folds
The importance of multivariate interactions
The importance of multivariate interactions
The importance of multivariate interactions
The importance of multivariate interactions

A: pattern

subject A

subject B
The importance of multivariate interactions

A: pattern

subject A

B: example data points

target

non-target

subject B
The importance of multivariate interactions
The importance of multivariate interactions
Error measures

Computing the accuracy is simple, just count how many examples you have classified correctly!
Error measures

Computing the accuracy is simple, just count how many examples you have classified correctly!

Yes, but …

What if the data is such that 99% of the samples are belonging to the non-target class. If I constantly predict non-target, this will be a good model.
Error measures

True positive rate (or sensitivity, recall):

\[ TPR = \frac{TP}{P} \]

True negative rate (or specificity)

\[ TNR = \frac{TN}{N} \]

False positive rate

\[ FPR = \frac{FP}{N} \]
Error measures: balanced accuracy

True positive rate (or sensitivity, recall):

$$TPR = \frac{TP}{P}$$

True negative rate (or specificity)

$$TNR = \frac{TN}{N}$$

Possible to combine TPR and TNR in a balanced accuracy by averaging.
Error measures: area under curve
Questions?
The hands on session (the work)
Data

- Visual ERP data (6x6) matrix speller

- 1:5 ratio of target to non-targets

- 15 iterations

- 12 stimuli per iteration

- 64 channels at 240 Hz
Find the target samples!
Feedback