Brain-Computer Interfacing
WS 2018/2019 – Lecture #12

Benjamin Blankertz
(based on material of Carmen Vidaurre)

Lehrstuhl für Neurotechnologie, TU Berlin

benjamin.blankertz@tu-berlin.de

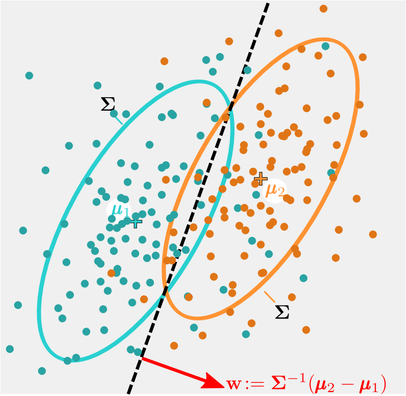
23 · Jan · 2019

Today's Topics

- Nonstationarities in BCI Data
- Adaptation of mean and covariance matrix
- Adaptation for the extended covariance matrix
- Supervised and unsupervised adaptation of LDA
- Critical issue in validation: block effects

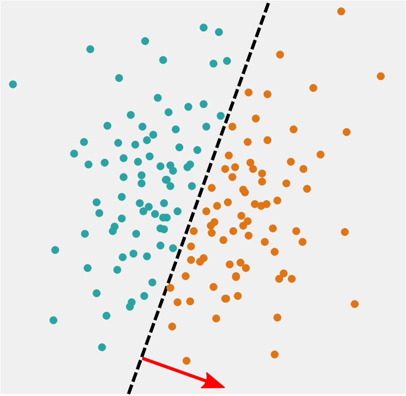
Recap: Classification (with LDA)

Calibration



estimate separation from
calibration data {samples, labels}

Application



estimate labels of incoming
samples using separation line

This approach relies on the **stationarity** assumption: samples during the application come from the same distribution as samples in the calibration.

Sources of Changes in EEG Signals

[black: cause of nonstationarity | blue: affected entity]

- Intended *Class related* short-term changes: performance of different mental tasks. Class means of the features
- *Class related* long-term changes: due to feedback training (learning). Class means of the features; maybe also common covariance
- *Class unrelated* mid/long-term changes: e.g., fatigue or lack of concentration. ERP: Common covariance of the features; ERD: Common mean of features + CSP filters become suboptimal
- Variation of other *noise sources*: e.g. changing impedance of the electrodes. ERP: Common covariance of the features; ERD: Common mean of features + CSP filters become suboptimal

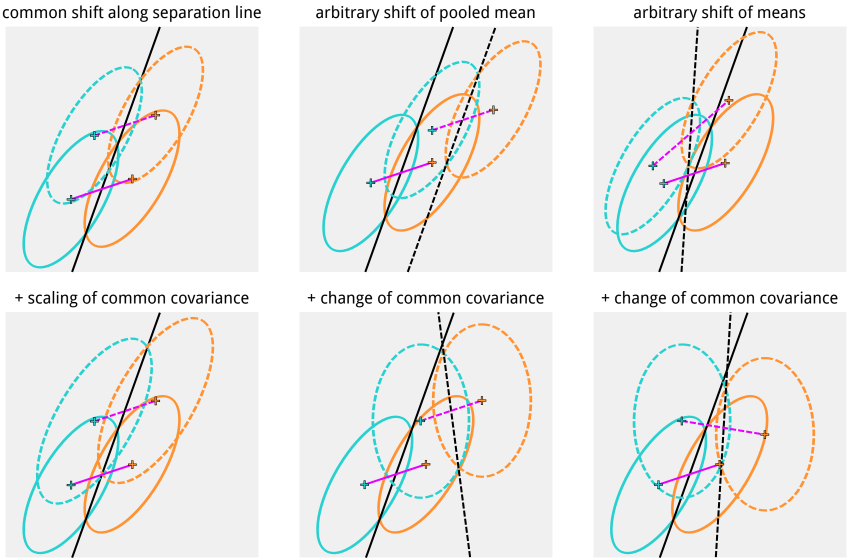
Some Definitions

Pooled Mean: is the mean over all samples (regardless of class affiliation). If the number of samples per class is the same, the pooled mean is equal to average of the classwise means $1/2(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$.

Pooled Covariance: is the covariance calculated across all samples (regardless of class affiliation).

It can be shown that the weight vector of ordinary LDA, $\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ coincides with $\boldsymbol{\Sigma}_{\text{pooled}}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ under the assumption that both classes have the same number of samples.

Welcome to the Zoo of Nonstationarities



no adaptation required | unsupervised adaptation works | supervised adapatation required

Adaptation of LDA

For an adaptive version of LDA, we reestimate mean and covariance matrices continuously during adaptation after each trial. To formalize this, we use k as an index for trials and write $\mathbf{x}(k)$ for the feature vector of trial k and $\hat{\boldsymbol{\mu}}(k)$, $\hat{\boldsymbol{\Sigma}}(k)$, ... to denote the estimate of mean and covariance matrix after having observed trial k. Note, that these need to be estimated for both classes.

A straight forward version of an adaptive LDA is to estimate mean and covariance from the last N number of trials and then to recalculate LDA:

$$\hat{\boldsymbol{\mu}}(k) = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{x}(k-n) \tag{1}$$

$$\hat{\boldsymbol{\Sigma}}(k) = \frac{1}{N-1} \sum_{n=0}^{N-1} (\mathbf{x}(k-n) - \hat{\boldsymbol{\mu}}(k))(\mathbf{x}(k-n) - \hat{\boldsymbol{\mu}}(k))^{\top} \tag{2}$$

Adaptive Estimation of the Mean

There is also a recursive version to estimate the mean:

$$\hat{\boldsymbol{\mu}}(k) = \hat{\boldsymbol{\mu}}(k-1) + \frac{1}{N}\left(\mathbf{x}(k) - \mathbf{x}(k-N)\right)$$

This approach has the disadvantage that N past samples have to be buffered, and that older samples have the same weight as recent ones.

Update rule with exponential weighting that does require no memory:

$$\hat{\boldsymbol{\mu}}(k) = (1-\alpha)\,\hat{\boldsymbol{\mu}}(k-1) \;+\; \alpha\,\mathbf{x}(k) \tag{3}$$

where α is the update coefficient for the adaptive mean adaptation.

The initial value $\hat{\boldsymbol{\mu}}(0)$ is the mean estimated from the calibration data.

Adaptive Estimation of the Covariance Matrix

Analog to the mean, we have an adaptive estimator of the covariance matrix:

$$\hat{\boldsymbol{\Sigma}}(k) := (1-\beta)\,\hat{\boldsymbol{\Sigma}}(k-1) \;+\; \beta\,(\mathbf{x}(k)-\hat{\boldsymbol{\mu}}(k))(\mathbf{x}(k)-\hat{\boldsymbol{\mu}}(k))^\top$$

where β is the update coefficient for the adaptive covariance estimation.

The initial value $\hat{\boldsymbol{\Sigma}}(0)$ is the covariance estimated from the calibration data.

Here, $\hat{\boldsymbol{\mu}}(k)$ would need to be estimated in parallel as described above. However, this can lead to adverse effects, in particular, if different update coefficients are chosen for $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$.

So, it is better to have a common adaptive estimation of mean and covariances. To that end, we will introduce the extended covariance matrix.

Extended Covariance Matrix

We define the **extended covariance matrix** (ECM) $\boldsymbol{E}$ as

$$\begin{aligned}\mathbf{E} &= \frac{1}{K}\sum_{k=1}^{K}[1;\mathbf{x}(k)]\,[1;\mathbf{x}(k)]^\top \\ &= \frac{1}{K}\sum_{k=1}^{K}\left[\begin{array}{c|c} 1 & \mathbf{x}(k)^\top \\ \hline \mathbf{x}(k) & \mathbf{x}(k)\mathbf{x}(k)^\top \end{array}\right] = \left[\begin{array}{c|c} 1 & \boldsymbol{\mu}^\top \\ \hline \boldsymbol{\mu} & \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top \end{array}\right]\end{aligned} \tag{4}$$

From the ECM $\mathbf{E}(k)$, the covariance matrix $\boldsymbol{\Sigma}$ as well as the mean $\boldsymbol{\mu}$ can be estimated (Matlab indexing notation):

- Mean: $\boldsymbol{\mu} = \mathbf{E}(2\text{:end}, 1)$
- Covariance matrix: $\boldsymbol{\Sigma} = \left(\mathbf{E}(2\text{:end},2\text{:end}) - \boldsymbol{\mu}\boldsymbol{\mu}^\top)\right)$.

Therefore, developing an update rule for the ECM will enable us derive consistent estimates of both, mean and covariance.

Adaptive Estimation of the Extended Covariance Matrix

Adaptive ECM estimator:

$$\mathbf{E}(k) = (1 - \beta)\, \mathbf{E}(k-1) \; + \; \beta\, [1; \mathbf{x}(k)]\, [1; \mathbf{x}(k)]^\top \tag{5}$$

Eq. (5) provides an efficient method to adaptively calculate $\mathbf{E}(k)$ (and thereby also $\mathbf{\Sigma}(k)$).

But for LDA we need the inverse $\mathbf{\Sigma}(k)^{-1}$, and inversion in the adaptive setting (i.e. after each trial k) would mean a considerable computational load.

Matrix Inversion Lemma

Now, we need a method to determine the inverse of the adaptive formula for the extended covariance matrix eq. (5). This is provided by the Matrix Inversion Lemma (Sherman-Morrison-Woodbury identity):

Given an invertible matrix $\mathbf{A}$ *in the form:*

$$\mathbf{A} = \mathbf{B} + \mathbf{UDV}$$

with invertible $\mathbf{B}$ *and* $\mathbf{D}$, *the inverse of* $\mathbf{A}$ *can be calculated in the following way:*

$$\begin{aligned} \mathbf{A}^{-1} &= (\mathbf{B} + \mathbf{UDV})^{-1} \\ &= \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{U}\left(\mathbf{D}^{-1} + \mathbf{VB}^{-1}\mathbf{U}\right)^{-1}\mathbf{VB}^{-1} \end{aligned} \tag{6}$$

Adaptive Estimation of Inverse Extended Covariance Matrix

We can apply the matrix inversion lemma to calculate the inverse $\mathbf{E}(k)^{-1}$ from eq. (5) by defining

$$\begin{aligned}\mathbf{A} &= \mathbf{E}(k)\\ \mathbf{B} &= (1-\beta)\mathbf{E}(k-1)\\ \mathbf{U} &= \mathbf{V}^\top = [1; \mathbf{x}(k)]\\ \mathbf{D} &= \beta\end{aligned}$$

and obtain with some standard matrix calculations and the abbreviations $\mathbf{E} := \mathbf{E}(k-1)$ and $\mathbf{u} = [1; \mathbf{x}(k)]$

$$\mathbf{E}(k)^{-1} = \frac{\mathbf{E}^{-1} - \frac{\beta}{1-\beta+\beta\cdot\mathbf{u}^\top\mathbf{E}^{-1}\mathbf{u}}\,\mathbf{E}^{-1}\mathbf{u}(\mathbf{E}^{-1}\mathbf{u})^\top}{1-\beta}. \qquad (7)$$

Importantly, $\mathbf{u}^\top\mathbf{E}^{-1}\mathbf{u}$ is a scalar. As a result, $\mathbf{E}(k)^{-1}$ can be calculated with simple matrix–vector multiplication and addition only. (Just $\mathbf{E}(0)^{-1}$ needs to be calculated initially.)

Extracting the Covariance from the Inverse of the ECM

Finally, we need to extract the inverse of the ordinary covariance matrix $\mathbf{\Sigma}$ from the inverse of $\mathbf{E}^{-1}$.

We use the rule for the inverse of a block matrix (with $\mathbf{S} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$)

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{B} \\ \hline \mathbf{C} & \mathbf{D} \end{array}\right]^{-1} = \left[\begin{array}{c|c} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\mathbf{S}^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}\mathbf{S}^{-1} \\ \hline -\mathbf{S}^{-1}\mathbf{C}\mathbf{A}^{-1} & \mathbf{S}^{-1} \end{array}\right] \tag{8}$$

to transform the inverse of ECM (note that $\mathbf{S} = \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top - \boldsymbol{\mu}\boldsymbol{\mu}^\top = \mathbf{\Sigma}$):

$$\begin{aligned} \mathbf{E}^{-1} &= \left[\begin{array}{c|c} 1 & \boldsymbol{\mu}^\top \\ \hline \boldsymbol{\mu} & \mathbf{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top \end{array}\right]^{-1} \\ &\overset{(8)}{=} \left[\begin{array}{c|c} 1 + \boldsymbol{\mu}^\top\mathbf{\Sigma}^{-1}\boldsymbol{\mu} & -\boldsymbol{\mu}^\top\mathbf{\Sigma}^{-1} \\ \hline -\mathbf{\Sigma}^{-1}\boldsymbol{\mu} & \mathbf{\Sigma}^{-1} \end{array}\right] \end{aligned} \tag{9}$$

This shows that we can easily extract $\mathbf{\Sigma}^{-1}$ from $\mathbf{E}^{-1}$ as a submatrix.

Practical Remarks

The inverse of the ECM can become asymmetric and singular. In order to avoid that correct the estimate by (sloppy formulation of overwriting $\mathbf{E}(k)^{-1}$):

$$\mathbf{E}(k)^{-1} = \frac{\left(\mathbf{E}(k)^{-1} + \mathbf{E}(k)^{-\top}\right)}{2}$$

Then, the inverse covariance matrix $\mathbf{\Sigma}^{-1}(k)$ can be obtained by adaptively estimating the extended covariance matrix with (7) and decomposing it according to equation (9):

$$\mathbf{\Sigma}^{-1}(k) = \mathbf{E}(k)^{-1}(2\text{:end},2\text{:end})$$

The mean is adaptively estimated with eqn (3), where a different update coefficient may be used.

Useful Adaptation Schemes

Pooled Mean: is the mean over all samples (regardless of class affiliation). If the number of samples per class is the same, the pooled mean is equal to average of the classwise means $1/2(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$.

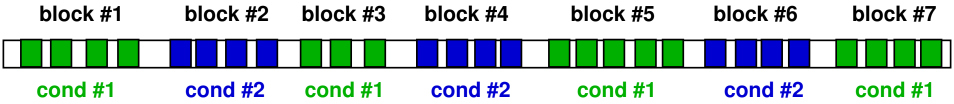
Pooled Covariance: is the covariance calculated across all samples (regardless of class affiliation).

- **PMean**: Unsupervised adaptation updating the *pooled mean* [Vidaurre et al, 2011a].
- **PMean-PCov**: Unsupervised adaptation updating the *pooled mean* and the *pooled covariance matrix* [Vidaurre et al, 2011a].
- **Mean-PCov**: Supervised adaptation updating the *class means* and the *pooled covariance matrix* [Vidaurre et al, 2011b]. Supervised adaptation of the common covariance matrix would also be possible here.

Part II

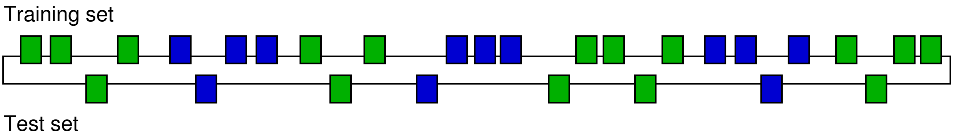
Block Design

Assume the task is to discriminate between mental states in different conditions. We say that an experiment has a block design, if the periods for which there is no alternation between conditions are longer than the intended change of states in online operation.

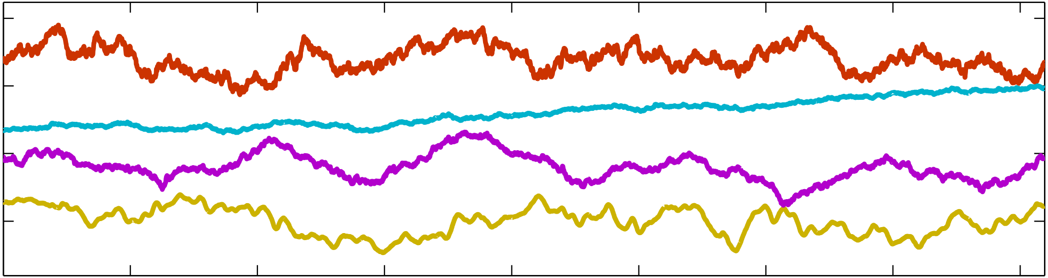


A problem arises, if the performance is estimated for such a data set by cross validation.

Slowly Changing Variables



Due to the autocorrelation of the EEG (many slowly changing variables of background activity), single-trials are not independent. For an ordinary cross validation in a block design dataset, the requirement of independence between training and test set is violated.



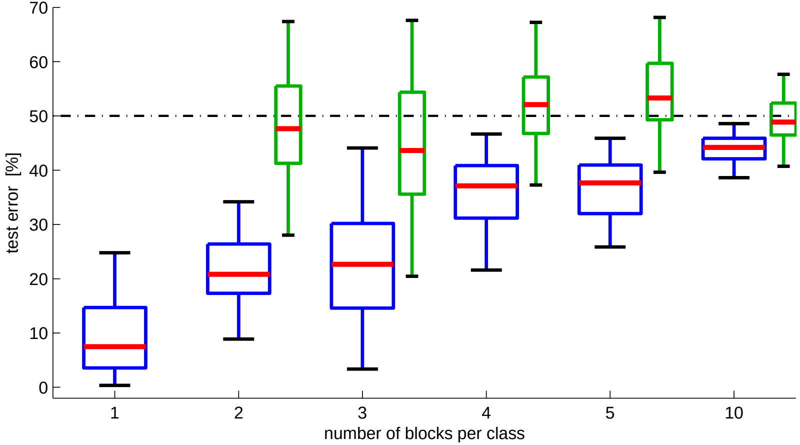
A Validation Test

To demonstrate impact of block design in cross validation, we perform cross validation in the following setting. Taking an arbitrary EEG data set, we assign **fake** labels (regardless of what happened during the recording) like this:

and so on.

Results of the Validation Test

From each block single-trials are extracted of length 1s. This procedure was performed for 80 EEG data sets. Blue boxplots show the results of cross-validation:



For comparison, results for **leave-one-block-out** validation are shown in green. $\Rightarrow$ In block design, cross-validation may underestimate the generalization error.

Hall of Shame in Single-Trial EEG Analysis (be aware!)

- preprocessing methods that use statistics of the whole data set like ICA, or normalization of features
  (particularly severe for methods that use label information like CSP)
- loss function not appropriate (e.g., unbalanced classes)
- artifacts/outliers are rejected from the whole data set (resulting in a simplified test set)
- features are selected on the whole data set, including trials that are later in the test set
- selection of parameters by cross validation on the whole data set and report the performance for the selected values
- non-stationarity of the data disregarded (chronological training / test data spilt vs. cross validation)
- insufficient validation for paradigms with block design

[Lemm et al, NeuroImage 2011]

Lessons Learnt

After this lecture you should

- be familar with supervised and unsupervised adaptation methods,
- in particular for updating the mean and the covariance matrix for LDA,
- and know how to implement them efficiently. Furthermore, you should
- be well aware of the issues in validating experiments with block design and
- know how to avoid them.

References I

- Lemm, S., Blankertz, B., Dickhaus, T., and Müller, K.-R. (2011).
  Introduction to machine learning for brain imaging.
  NeuroImage, 56:387–399.
- Vidaurre, C., Kawanabe, M., von Bünau, P., Blankertz, B., and Müller, K.-R. (2011a).
  Toward unsupervised adaptation of lda for brain-computer interfaces.
  IEEE Trans Biomed Eng, 58(3):587 –597.
- Vidaurre, C., Sannelli, C., Müller, K.-R., and Blankertz, B. (2011b).
  Co-adaptive calibration to improve BCI efficiency.
  J Neural Eng, 8(2):025009 (8pp).
- Vidaurre, C., Sannelli, C., Müller, K.-R., and Blankertz, B. (2011c).
  Machine-learning based co-adaptive calibration.
  Neural Comput, 23(3):791–816.

$$
\begin{aligned}
\boldsymbol{\Sigma} &= \frac{1}{K}\sum_{k=1}^{K}(\mathbf{x}(k)-\boldsymbol{\mu})(\mathbf{x}(k)-\boldsymbol{\mu})^{\top} && (10)\\
&= \frac{1}{K}\sum_{k=1}^{K}\left(\mathbf{x}(k)\mathbf{x}(k)^{\top}-\mathbf{x}(k)\boldsymbol{\mu}^{\top}-\boldsymbol{\mu}\mathbf{x}(k)^{\top}+\boldsymbol{\mu}\boldsymbol{\mu}^{\top}\right) && (11)\\
&= \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}(k)\mathbf{x}(k)^{\top}+\left(\frac{1}{K}\sum_{k=1}^{K}\mathbf{x}(k)\right)\boldsymbol{\mu}^{\top}-\boldsymbol{\mu}\left(\frac{1}{K}\sum_{k=1}^{K}\mathbf{x}(k)^{\top}\right)+\boldsymbol{\mu}\boldsymbol{\mu}^{\top} && (12)\\
&= \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}(k)\mathbf{x}(k)^{\top}-\boldsymbol{\mu}\boldsymbol{\mu}^{\top}-\boldsymbol{\mu}\boldsymbol{\mu}^{\top}+\boldsymbol{\mu}\boldsymbol{\mu}^{\top} && (13)\\
&= \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}(k)\mathbf{x}(k)^{\top}-\boldsymbol{\mu}\boldsymbol{\mu}^{\top} && (14)
\end{aligned}
$$