Stationary Subspace Analysis

Paul von Bünau Frank C. Meinecke Klaus-R. Müller

Machine Learning Group, Computer Science Dept., TU Berlin

Algebraic Methods Workshop at NIPS*08

Outline

Motivation

- Non-stationarities can be found in many real-world data, yet they challenge standard Machine Learning methods.
- Different training and test distributions:
  $\longrightarrow$ Problems to generalise.

Motivation

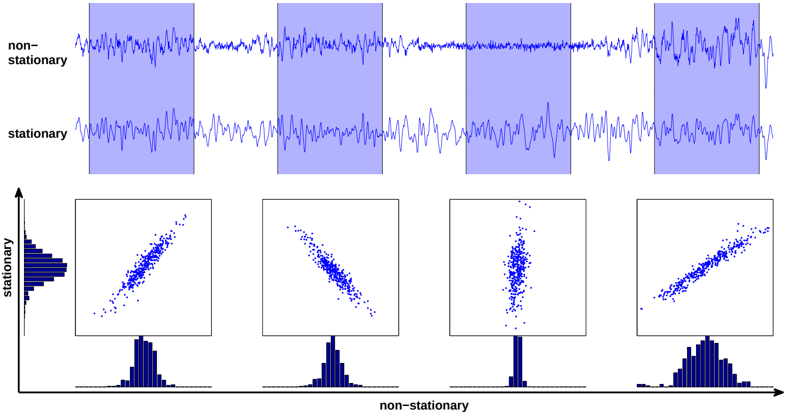
- Non-stationarities can be found in many real-world data, yet they challenge standard Machine Learning methods.
- Different training and test distributions:
  $\longrightarrow$ Problems to generalise.

Observation:

Data generating systems are often only partly non-stationary.

- Getting rid of the non-stationary part might help.
- Understanding the nature of the non-stationarity is an interesting endeavour in its own right.

Stationary and Non-stationary subspaces



Generative Model

Assumption

The non-stationarity is confined to a linear subspace of the D-dimensional data space.

Generative Model

Assumption

The non-stationarity is confined to a linear subspace of the D-dimensional data space.

- d stationary source signals $s^{\mathfrak{s}}(t) \in \mathbb{R}^d$

Generative Model

Assumption

The non-stationarity is confined to a linear subspace of the D-dimensional data space.

- d stationary source signals $s^{\mathfrak{s}}(t) \in \mathbb{R}^d$
- $D - d$ non-stationary source signals $s^{\mathfrak{n}}(t) \in \mathbb{R}^{(D-d)}$

Generative Model

Assumption

The non-stationarity is confined to a linear subspace of the D-dimensional data space.

- d stationary source signals $s^{\mathfrak{s}}(t) \in \mathbb{R}^d$
- $D - d$ non-stationary source signals $s^{\mathfrak{n}}(t) \in \mathbb{R}^{(D-d)}$
- Observed signals: instantaneous linear superpositions of sources

$$x(t) = As(t) = \begin{bmatrix} A^{\mathfrak{s}} & A^{\mathfrak{n}} \end{bmatrix} \begin{bmatrix} s^{\mathfrak{s}}(t) \\ s^{\mathfrak{n}}(t) \end{bmatrix}$$

Aim of Stationary Subspace Analysis

$$x(t) = As(t) = \begin{bmatrix} A^{\mathfrak{s}} & A^{\mathfrak{n}} \end{bmatrix} \begin{bmatrix} s^{\mathfrak{s}}(t) \\ s^{\mathfrak{n}}(t) \end{bmatrix}$$

Goal

Given only $x(t)$, find an estimate $\hat{A}$ for the mixing matrix, such that $\hat{B} = \hat{A}^{-1}$ separates $\mathfrak{s}$-sources from $\mathfrak{n}$-sources.

$$\begin{bmatrix} \hat{s}^{\mathfrak{s}}(t) \\ \hat{s}^{\mathfrak{n}}(t) \end{bmatrix} = \hat{B}x(t) = \begin{bmatrix} \hat{B}^{\mathfrak{s}} \\ \hat{B}^{\mathfrak{n}} \end{bmatrix} x(t)$$

Aim of Stationary Subspace Analysis

$$x(t) = As(t) = \begin{bmatrix} A^{\mathfrak{s}} & A^{\mathfrak{n}} \end{bmatrix} \begin{bmatrix} s^{\mathfrak{s}}(t) \\ s^{\mathfrak{n}}(t) \end{bmatrix}$$

Goal

Given only $x(t)$, find an estimate $\hat{A}$ for the mixing matrix, such that $\hat{B} = \hat{A}^{-1}$ separates $\mathfrak{s}$-sources from $\mathfrak{n}$-sources.

$$\begin{bmatrix} \hat{s}^{\mathfrak{s}}(t) \\ \hat{s}^{\mathfrak{n}}(t) \end{bmatrix} = \hat{B}x(t) = \begin{bmatrix} \hat{B}^{\mathfrak{s}} \\ \hat{B}^{\mathfrak{n}} \end{bmatrix} x(t)$$

Clearly, $\hat{A} = A$ is a solution. But are there other solutions?

Symmetries and Invariances

Let's express the true $A^{\mathfrak{s}}$ and $A^{\mathfrak{n}}$ as linear combinations of the respective estimated subspaces

$$
\begin{aligned}
A^{\mathfrak{s}} &= \hat{A}^{\mathfrak{s}} M_1 + \hat{A}^{\mathfrak{n}} M_2 \\
A^{\mathfrak{n}} &= \hat{A}^{\mathfrak{s}} M_3 + \hat{A}^{\mathfrak{n}} M_4
\end{aligned}
$$

Symmetries and Invariances

Let's express the true $A^{\mathfrak{s}}$ and $A^{\mathfrak{n}}$ as linear combinations of the respective estimated subspaces

$$\begin{aligned} A^{\mathfrak{s}} &= \hat{A}^{\mathfrak{s}} M_1 + \hat{A}^{\mathfrak{n}} M_2 \\ A^{\mathfrak{n}} &= \hat{A}^{\mathfrak{s}} M_3 + \hat{A}^{\mathfrak{n}} M_4 \end{aligned}$$

The composite transformation (true mixing followed by the estimated demixing) reads

$$\begin{bmatrix} \hat{s}^{\mathfrak{s}}(t) \\ \hat{s}^{\mathfrak{n}}(t) \end{bmatrix} = \hat{B} A s(t) = \begin{bmatrix} \hat{B}^{\mathfrak{s}} A^{\mathfrak{s}} & \hat{B}^{\mathfrak{s}} A^{\mathfrak{n}} \\ \hat{B}^{\mathfrak{n}} A^{\mathfrak{s}} & \hat{B}^{\mathfrak{n}} A^{\mathfrak{n}} \end{bmatrix} s(t) = \begin{bmatrix} M_1 & M_3 \\ M_2 & M_4 \end{bmatrix} \begin{bmatrix} s^{\mathfrak{s}}(t) \\ s^{\mathfrak{n}}(t) \end{bmatrix}$$

Symmetries and Invariances

Let's express the true $A^{\mathfrak{s}}$ and $A^{\mathfrak{n}}$ as linear combinations of the respective estimated subspaces

$$\begin{aligned} A^{\mathfrak{s}} &= \hat{A}^{\mathfrak{s}} M_1 + \hat{A}^{\mathfrak{n}} M_2 \\ A^{\mathfrak{n}} &= \hat{A}^{\mathfrak{n}} M_4 \end{aligned}$$

The composite transformation (true mixing followed by the estimated demixing) reads

$$\begin{bmatrix} \hat{s}^{\mathfrak{s}}(t) \\ \hat{s}^{\mathfrak{n}}(t) \end{bmatrix} = \hat{B} A s(t) = \begin{bmatrix} \hat{B}^{\mathfrak{s}} A^{\mathfrak{s}} & \hat{B}^{\mathfrak{s}} A^{\mathfrak{n}} \\ \hat{B}^{\mathfrak{n}} A^{\mathfrak{s}} & \hat{B}^{\mathfrak{n}} A^{\mathfrak{n}} \end{bmatrix} s(t) = \begin{bmatrix} M_1 & 0 \\ M_2 & M_4 \end{bmatrix} \begin{bmatrix} s^{\mathfrak{s}}(t) \\ s^{\mathfrak{n}}(t) \end{bmatrix}$$

Restriction to orthogonal demixing matrices

- Since M_1, M_2, M_4 are arbitrary, $A^{\mathfrak{s}}$ can always be chosen such that it is orthogonal to $A^{\mathfrak{n}}$.

Restriction to orthogonal demixing matrices

- Since M_1, M_2, M_4 are arbitrary, $A^{\mathfrak{s}}$ can always be chosen such that it is orthogonal to $A^{\mathfrak{n}}$.
- Chosing orthogonal bases within each of these estimated subspaces, we have effectively restricted ourselves to the estimation of an orthogonal mixing matrix.

Restriction to orthogonal demixing matrices

- Since M_1, M_2, M_4 are arbitrary, $A^{\mathfrak{s}}$ can always be chosen such that it is orthogonal to $A^{\mathfrak{n}}$.
- Chosing orthogonal bases within each of these estimated subspaces, we have effectively restricted ourselves to the estimation of an orthogonal mixing matrix.

Result

We can restrict our search for the mixing matrix to the space of orthogonal matrices even if the model allows general (i.e. non-orthogonal) mixing matrices.

Measuring (Non-)Stationarity

Stationarity

Given N data sets, we will consider a set of d estimated sources as stationary, if the joint distribution of these sources stays the same.

Measuring (Non-)Stationarity

Stationarity

Given N data sets, we will consider a set of d estimated sources as stationary, if the joint distribution of these sources stays the same.

Objective Function

Pairwise Kullback-Leibler divergence between the distributions of the projected data (using $\hat{B}^{\mathfrak{s}}$)

Measuring (Non-)Stationarity

Stationarity

Given N data sets, we will consider a set of d estimated sources as stationary, if the joint distribution of these sources stays the same.

Objective Function

Pairwise Kullback-Leibler divergence between the distributions of the projected data (using $\hat{B}^{\mathfrak{s}}$)

Gaussian Approximation

Consider only differences in the first two moments
$\longrightarrow$ KL-Divergence between Gaussians (max. Entropy principle)

The Optimization Problem

To stay on the manifold of orthogonal matrices: multiplicative updates with rotation matrices ($RR^\top = I$).

$$\hat{B}^{\text{start}} = I \qquad\qquad \hat{B}^{\text{new}} \leftarrow R\hat{B}$$

The Optimization Problem

To stay on the manifold of orthogonal matrices: multiplicative updates with rotation matrices ($RR^\top = I$).

$$\hat{B}^{\text{start}} = I \qquad \hat{B}^{\text{new}} \leftarrow R\hat{B}$$

The loss function

$$L_B(R) = \sum_{i<j} \mathrm{KL}\left[\mathcal{N}(\hat{\mu}_i^{\mathfrak{s}}, \hat{\Sigma}_i^{\mathfrak{s}}) \,||\, \mathcal{N}(\hat{\mu}_j^{\mathfrak{s}}, \hat{\Sigma}_j^{\mathfrak{s}})\right]$$

with

$$\hat{\mu}_i^{\mathfrak{s}} = I^d RB\hat{\mu}_i \quad \text{and} \quad \hat{\Sigma}_i^{\mathfrak{s}} = I^d RB\hat{\Sigma}_i (I^d RB)^\top$$

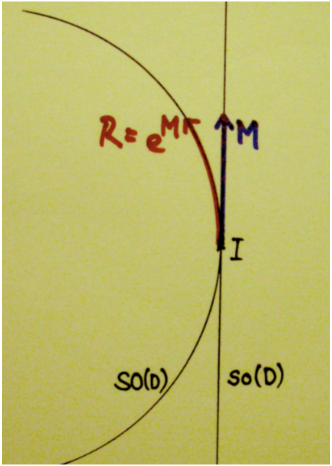
denoting estimated mean and covariance of the i-*th* data set projected to the $\mathfrak{s}$-subspace and $I^d \in \mathbb{R}^{d\times D}$ the identity matrix truncated to the first d rows.

Optimization in the Special Orthogonal Group

Manifold of all D-dimensional rotations:
Special Orthogonal Group SO(D).

Optimization in the Special Orthogonal Group

Manifold of all D-dimensional rotations:
Special Orthogonal Group SO(D).

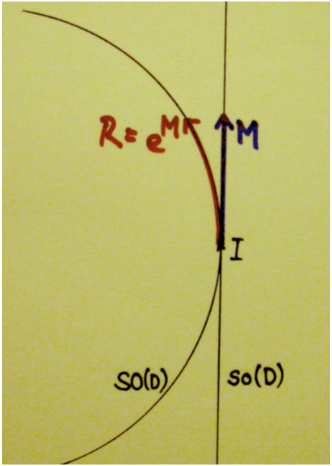


Optimization in the Special Orthogonal Group

Manifold of all D-dimensional rotations:
Special Orthogonal Group SO(D).

From Group Theory:

Every element of a Lie Group can be expressed as the exponential of an element from the corresponding Lie Algebra. (tangent space at I).



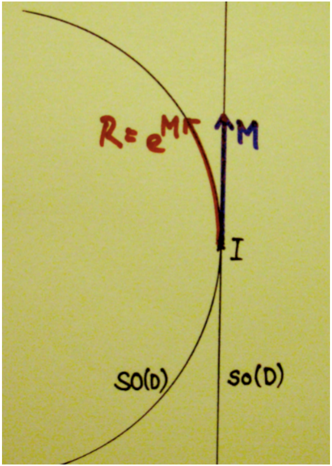
Optimization in the Special Orthogonal Group

Manifold of all D-dimensional rotations:
Special Orthogonal Group SO(D).

From Group Theory:

Every element of a Lie Group can be expressed as the exponential of an element from the corresponding Lie Algebra. (tangent space at I).

Linear space of all skew-symmetric matrices $M^\top = -M$:
Special Orthogonal Algebra $\mathfrak{so}$(D).



Optimization in the Special Orthogonal Group

We express R as

$$R = \exp(M)$$

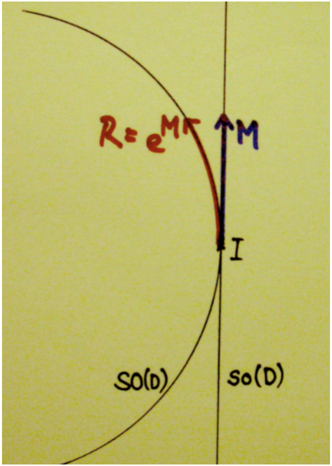
with $M^\top = -M$ and optimize the objective L_B in terms of M.

Interpretation of M_{ij}:

Angle of rotation of axis i towards axis j

The gradient translates to:

$$\left.\frac{\partial L_B}{\partial M}\right|_{M=0} = \left(\frac{\partial L_B}{\partial R}\right) R^\top - R \left(\frac{\partial L_B}{\partial R}\right)^\top$$



Optimization in the Special Orthogonal Group

Thus the gradient has the shape

$$\left.\frac{\partial L_B}{\partial M}\right|_{M=0} = \begin{bmatrix} 0 & Z \\ -Z^\top & 0 \end{bmatrix}$$

Z corresponds to rotations between $\mathfrak{s}$- and $\mathfrak{n}$-space.
Rotations within the two spaces do not change the objective.

Optimization in the Special Orthogonal Group

Thus the gradient has the shape

$$\left.\frac{\partial L_B}{\partial M}\right|_{M=0} = \begin{bmatrix} 0 & Z \\ -Z^\top & 0 \end{bmatrix}$$

Z corresponds to rotations between $\mathfrak{s}$- and $\mathfrak{n}$-space.
Rotations within the two spaces do not change the objective.

Result

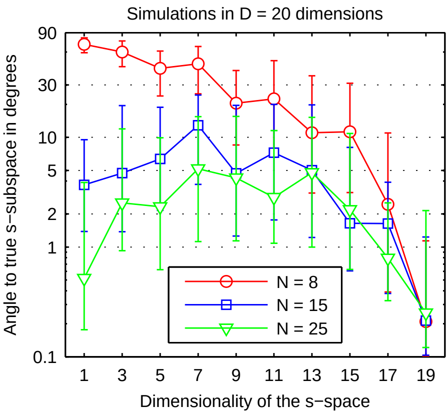
The number of variables is reduced to $d(D-d)$.

Simulations

Experimental Setup

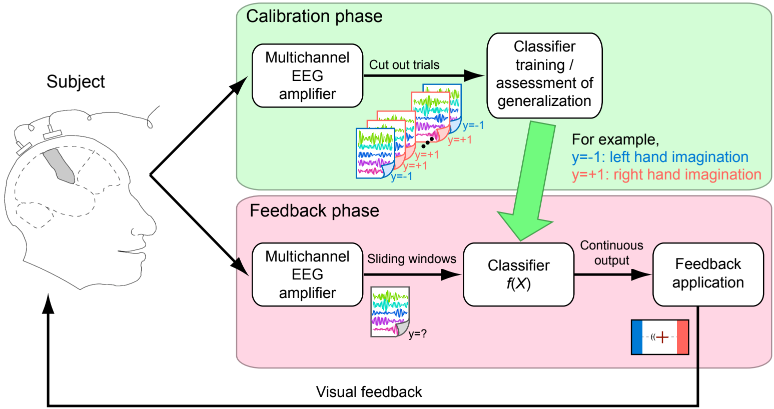
- N covariance matrices and means are sampled that are stationary in the first d coordinates.
- To each mean and covariance the same randomly sampled mixing matrix is applied.
- SSA is applied.
- The accuracy is measured as angle between the estimated $\mathfrak{n}$-subspace and the ground truth.

Simulations



- Input space dimension $D = 20$
- Number of data sets $N = 8, 15, 25$
- Performance as median angle to the true subspace
- 100 repetitions, error bars 25% to 75% quantile

BCI Experiment



BCI Experiment

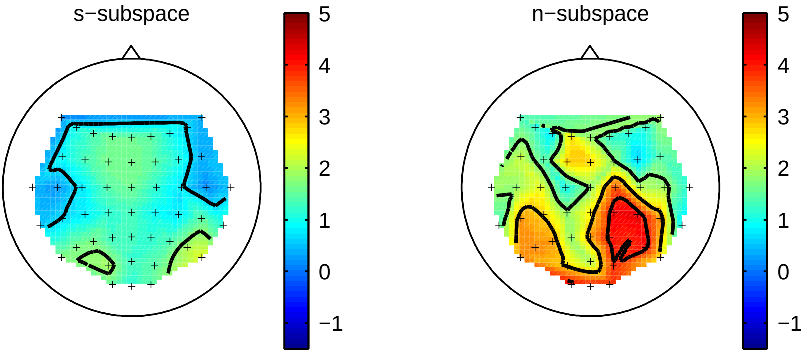
We induce changes in the strength of the α-rhythm by extracting it from a separate artefact measurement session (using ICA) and superimpose it on the data (adaptation and test set) in varying strengths.

BCI Experiment

We induce changes in the strength of the α-rhythm by extracting it from a separate artefact measurement session (using ICA) and superimpose it on the data (adaptation and test set) in varying strengths.

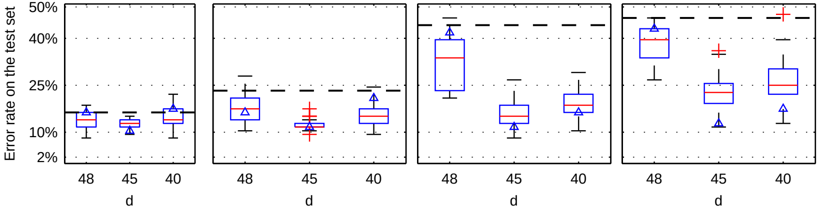
- Divide Data into 3 parts:
  1. Training Set, used for running SSA, to train the classifier
  2. Adaptation Set, used for running SSA
  3. Test Set, used for evaluating the classifier
- Estimate $\hat{A}$ over the training and adaptation part
- Train Classifier (CSP/LDA) within the $\mathfrak{s}$-space in training set
- Performance: misclassification rate on the test set

BCI Experiment



Relative power differences between training and test set.

BCI Experiment



- Boxplots show distribution of the test error rates
- Dashed black line: Test error rate of the baseline method (using all data).
- Blue triangle: error rate on the subspace with minimum objective function value

Conclusion

- We have presented an algorithm for decomposing a multivariate time-series into a stationary and a non-stationary component.
- We can restrict the search space to orthogonal transformations without limiting the applicability.
- Exploiting the underlying Lie-Group structure reduces the number of parameters and allows a stable and efficient optimization.
- Application to simulated and BCI data indicate that projecting out the $\mathfrak{n}$-sources can improve classification performance.

Thank You.