Machine-Learning Methods for Decoding Intentional Brain States

Jeremy Hill

MAX-PLANCK-GESELLSCHAFT

Max Planck Institute
for Biological Cybernetics

Tübingen, Germany

BIOLOGISCHE KYBERNETIK

BCI as a Potential Assistive Technology

- Complete paralysis (e.g. late-stage Amyotrophic Lateral Sclerosis)

BCI as a Potential Assistive Technology

- Complete paralysis (e.g. late-stage Amyotrophic Lateral Sclerosis)
  - Communication

BCI as a Potential Assistive Technology

- Complete paralysis (e.g. late-stage Amyotrophic Lateral Sclerosis)
  - Communication
- Disconnection of motor pathways (e.g. subcortical stroke, amputation)

BCI as a Potential Assistive Technology

- Complete paralysis (e.g. late-stage Amyotrophic Lateral Sclerosis)
    - Communication
- Disconnection of motor pathways (e.g. subcortical stroke, amputation)
    - Rehabilitation of movement

BCI as a Potential Assistive Technology

- Complete paralysis (e.g. late-stage Amyotrophic Lateral Sclerosis)
  - Communication
- Disconnection of motor pathways (e.g. subcortical stroke, amputation)
  - Rehabilitation of movement
  - Relief of phantom-limb pain

BCI as a Potential Assistive Technology

- Complete paralysis (e.g. late-stage Amyotrophic Lateral Sclerosis)
  - Communication
- Disconnection of motor pathways (e.g. subcortical stroke, amputation)
  - Rehabilitation of movement
  - Relief of phantom-limb pain
  - Control of prosthetics or FES

BCI as a Potential Assistive Technology

- Complete paralysis (e.g. late-stage Amyotrophic Lateral Sclerosis)
  - Communication
- Disconnection of motor pathways (e.g. subcortical stroke, amputation)
  - Rehabilitation of movement
  - Relief of phantom-limb pain
  - Control of prosthetics or FES
- Other...

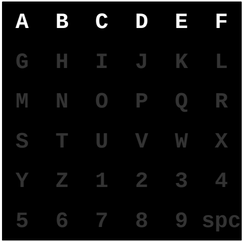
Induction

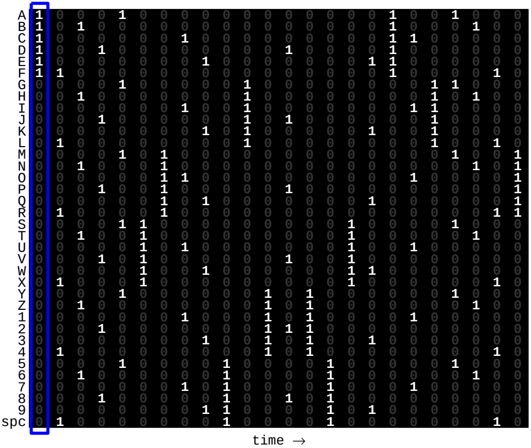
- Attention (overt and/or covert) to one of a number of stimuli

Induction

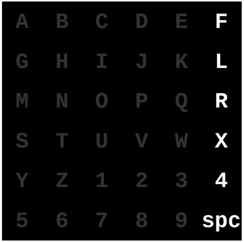
- Attention (overt and/or covert) to one of a number of stimuli
  - Most common example: visual grid speller (Farwell & Donchin 1988)

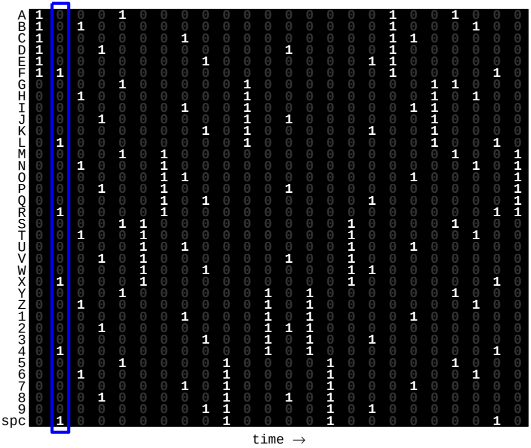
Induction


A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z 1 2 3 4
5 6 7 8 9 spc

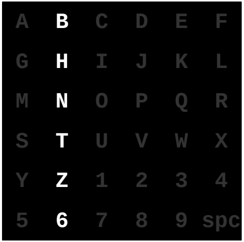

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9 spc
time →

Induction

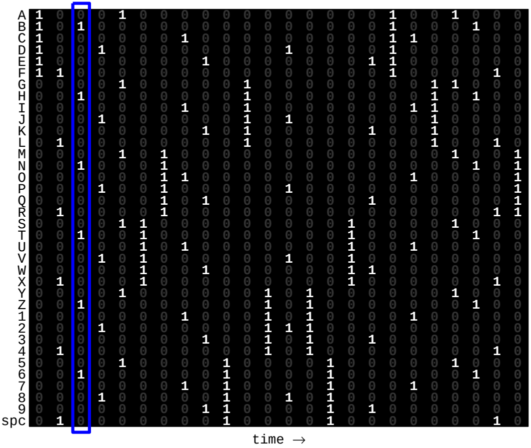

A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z 1 2 3 4
5 6 7 8 9 spc


A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9 spc
time →

Induction

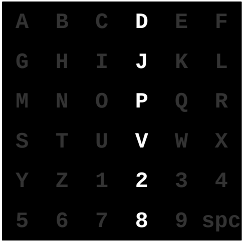


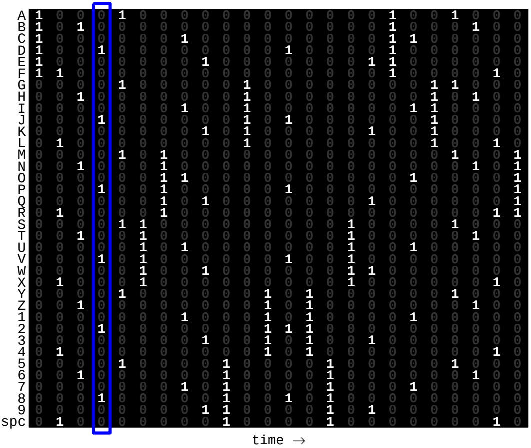
A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z 1 2 3 4
5 6 7 8 9 spc



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9 spc
time →

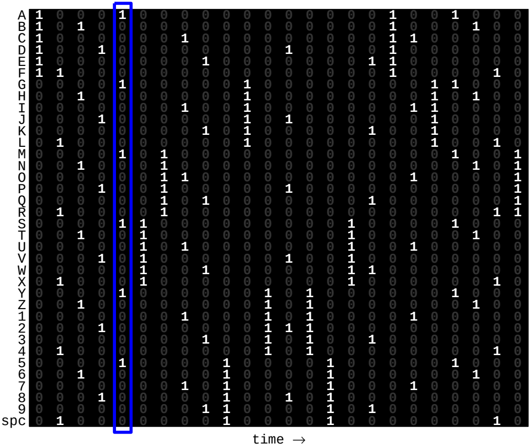
Induction


A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z 1 2 3 4
5 6 7 8 9 spc


A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9 spc
time →

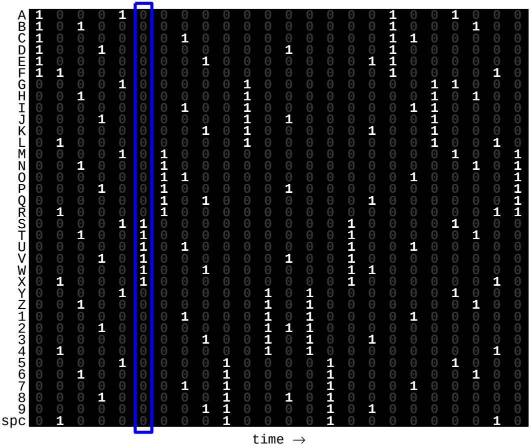
Induction


A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z 1 2 3 4
5 6 7 8 9 spc


A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9 spc
time →

Induction


A B C D E F
G H I J K L
M N O P Q R
S T U V W X
Y Z 1 2 3 4
5 6 7 8 9 spc


A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 1 2 3 4 5 6 7 8 9 spc
time →

Induction

- Attention (overt and/or covert) to one of a number of stimuli
  - Most common example: visual grid speller (Farwell & Donchin 1988)
  - BUT: for *completely* paralysed users, vision deteriorates.

Induction

- Attention (overt and/or covert) to one of a number of stimuli
  - Most common example: visual grid speller (Farwell & Donchin 1988)
  - BUT: for *completely* paralysed users, vision deteriorates.
    $\rightsquigarrow$ incentive to design auditory-/tactile-based methods.

Induction

- Attention (overt and/or covert) to one of a number of stimuli
  - Most common example: visual grid speller (Farwell & Donchin 1988)
  - BUT: for *completely* paralysed users, vision deteriorates.
    $\rightsquigarrow$ incentive to design auditory-/tactile-based methods.
- "Mental tasks"

Induction

- Attention (overt and/or covert) to one of a number of stimuli
  - Most common example: visual grid speller (Farwell & Donchin 1988)
  - BUT: for *completely* paralysed users, vision deteriorates.
    $\rightsquigarrow$ incentive to design auditory-/tactile-based methods.
- "Mental tasks"
  - Most common example: imagined movement of hands or feet.

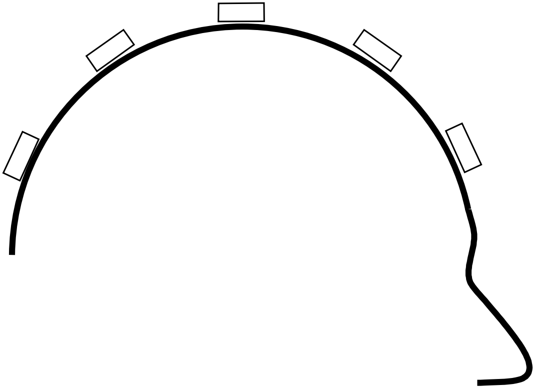
Induction

- Attention (overt and/or covert) to one of a number of stimuli
  - Most common example: visual grid speller (Farwell & Donchin 1988)
  - BUT: for *completely* paralysed users, vision deteriorates.
    $\rightsquigarrow$ incentive to design auditory-/tactile-based methods.
- "Mental tasks"
  - Most common example: imagined movement of hands or feet.
  - BUT: for users with motor-neuron disease, will the motor system continue functioning well enough long-term?

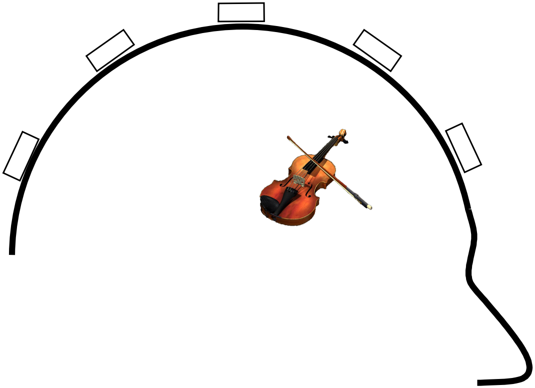
Induction

- Attention (overt and/or covert) to one of a number of stimuli
  - Most common example: visual grid speller (Farwell & Donchin 1988)
  - BUT: for *completely* paralysed users, vision deteriorates.
    $\rightsquigarrow$ incentive to design auditory-/tactile-based methods.
- "Mental tasks"
  - Most common example: imagined movement of hands or feet.
  - BUT: for users with motor-neuron disease, will the motor system continue functioning well enough long-term?
    $\rightsquigarrow$ incentive to explore non-motor mental tasks (e.g. covert visual attention without a specific target).

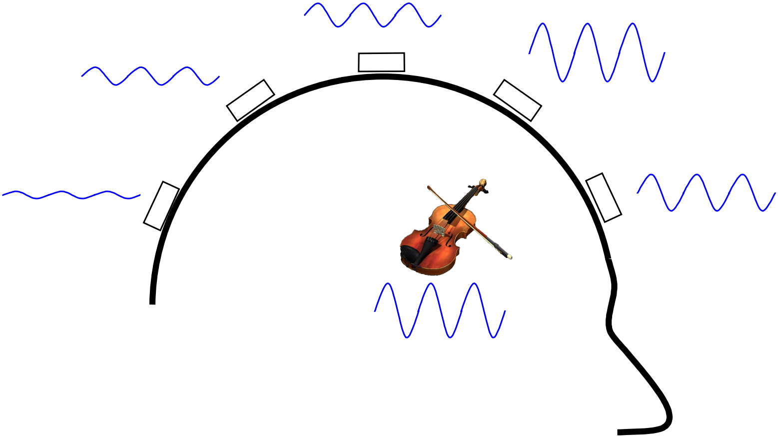
Why (and when) volume conduction matters



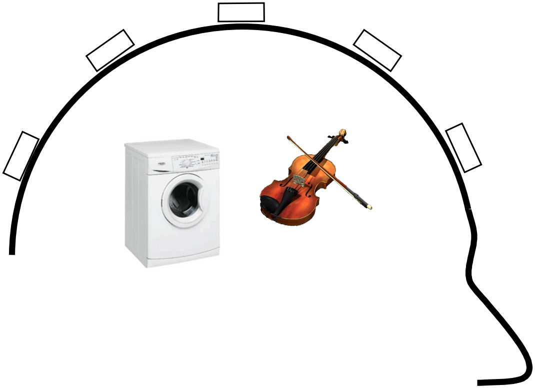
Why (and when) volume conduction matters



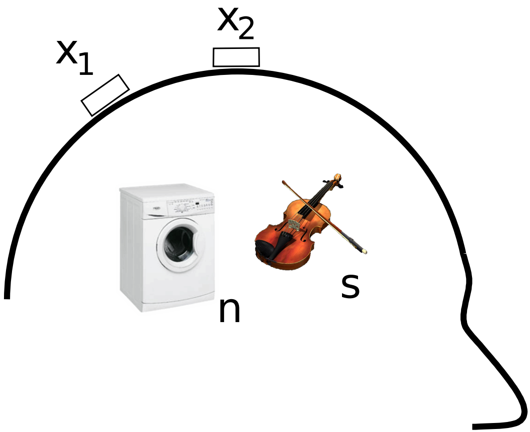
Why (and when) volume conduction matters



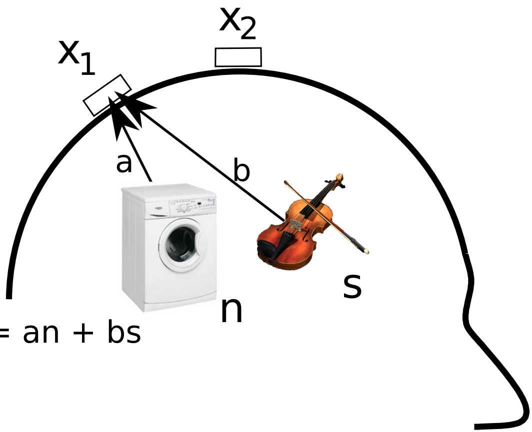
Why (and when) volume conduction matters



Why (and when) volume conduction matters

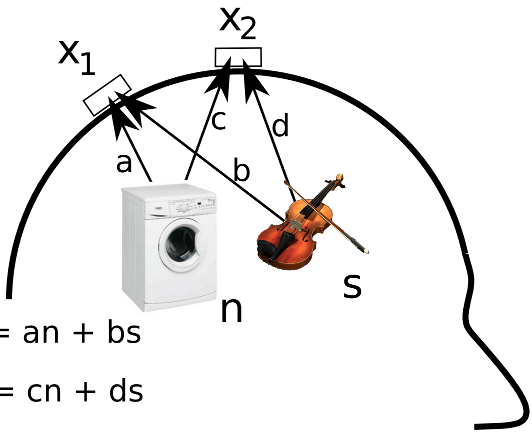


Why (and when) volume conduction matters



$x_1 = an + bs$

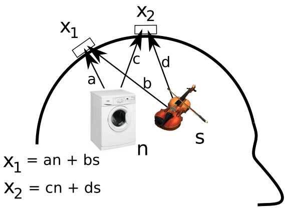
Why (and when) volume conduction matters

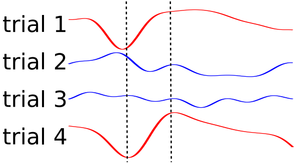


$$x_1 = an + bs$$

$$x_2 = cn + ds$$

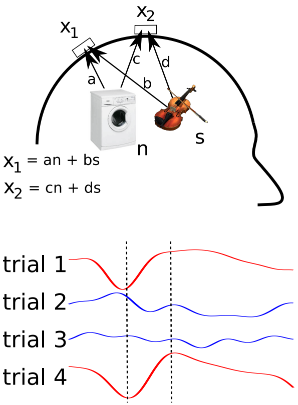
Why (and when) volume conduction matters





Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

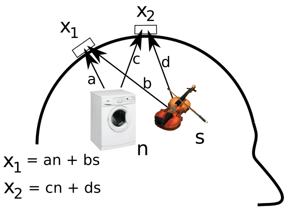
Why (and when) volume conduction matters

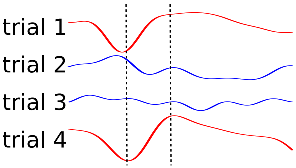


Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

$$f(\mathrm{X}) = w_1 x_1 \qquad + \quad w_2 x_2$$

Why (and when) volume conduction matters

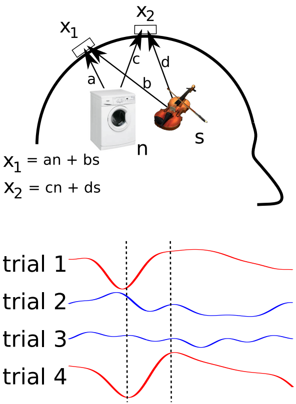




Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

$$f(\mathrm{X}) \quad = \quad w_1(an + bs) \quad + \quad w_2(cn + ds)$$

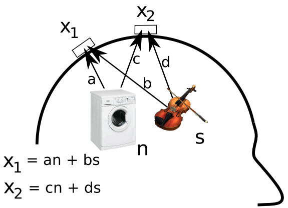
Why (and when) volume conduction matters

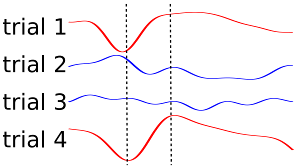


Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

$$f(\mathrm{X}) \quad = \quad w_1 an + w_1 bs + w_2 cn + w_2 ds$$

Why (and when) volume conduction matters

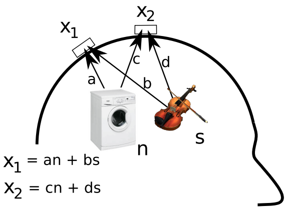


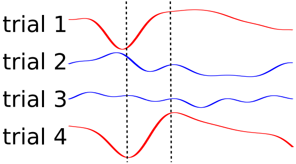


Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

$$f(\mathrm{X}) \quad = \quad (w_1 a + w_2 c)n + (w_1 b + w_2 d)s$$

Why (and when) volume conduction matters



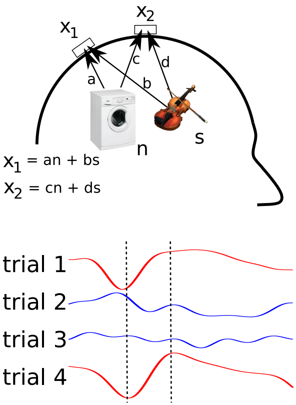


Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

$$f(\mathrm{X}) \;=\; (w_1 a + w_2 c)n + (w_1 b + w_2 d)s$$

If the classifier is good enough, it might be able to find a solution that isolates the signal entirely:

Why (and when) volume conduction matters



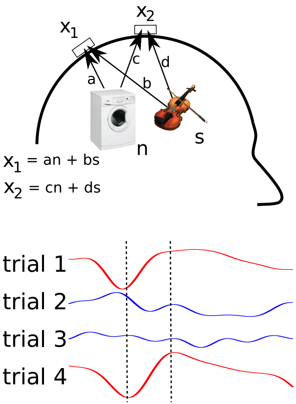
Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

$$f(\mathrm{X}) \quad = \quad (w_1 a + w_2 c)n + (w_1 b + w_2 d)s$$

If the classifier is good enough, it might be able to find a solution that isolates the signal entirely:

for example: $\quad w_1 = c, \quad w_2 = -a$

Why (and when) volume conduction matters



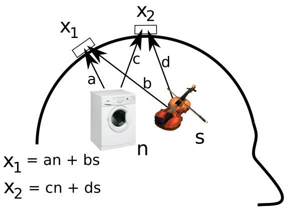
Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

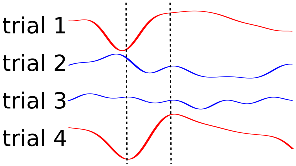
$$f(\mathrm{X}) \quad = \quad (ca - ac)n + (cb - ad)s$$

If the classifier is good enough, it might be able to find a solution that isolates the signal entirely:

for example: $w_1 = c, \quad w_2 = -a$

Why (and when) volume conduction matters





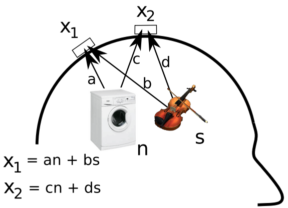
Case 1 (time-locked signals): Apply a linear classifier to (some *linear* transformation of) the raw data $\mathrm{X} = \{x_1, x_2\}$:

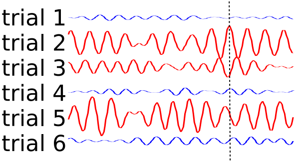
$$f(\mathrm{X}) \quad = \quad (cb - ad)s$$

If the classifier is good enough, it might be able to find a solution that isolates the signal entirely:

$$\text{for example:} \quad w_1 = c, \quad w_2 = -a$$

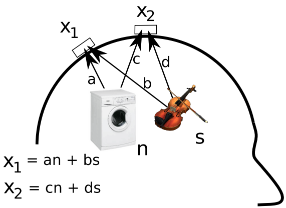
Why (and when) volume conduction matters

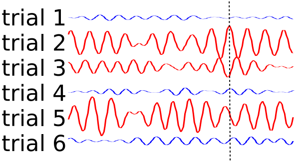




Case 2 (non-time-locked signals): Apply a linear classifier to some *non*-linear transformation of the raw data, for example power $\{x_1^2, x_2^2\}$:

Why (and when) volume conduction matters

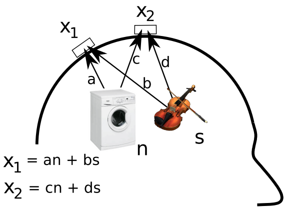


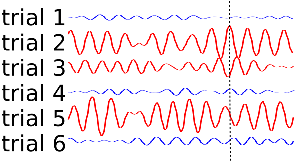


Case 2 (non-time-locked signals): Apply a linear classifier to some *non*-linear transformation of the raw data, for example power $\{x_1^2, x_2^2\}$:

$$f(\mathrm{X}) = w_1 x_1^2 + w_2 x_2^2$$

Why (and when) volume conduction matters

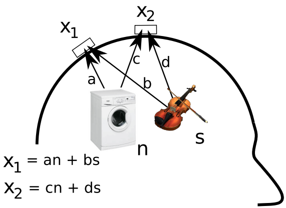


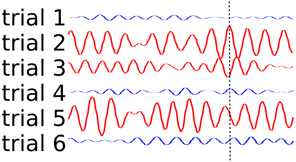


Case 2 (non-time-locked signals): Apply a linear classifier to some *non*-linear transformation of the raw data, for example power $\{x_1^2, x_2^2\}$:

$$f(\mathrm{X}) \;=\; w_1(an+bs)^2 \;+\; w_2(cn+ds)^2$$

Why (and when) volume conduction matters

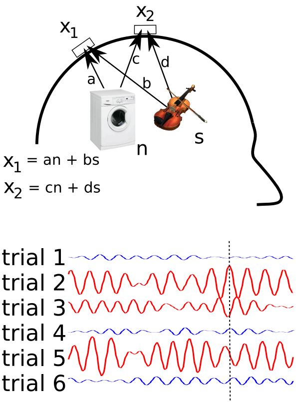




Case 2 (non-time-locked signals): Apply a linear classifier to some *non*-linear transformation of the raw data, for example power $\{x_1^2, x_2^2\}$:

$$f(\mathrm{X}) = w_1 a^2 n^2 + w_1 b^2 s^2 + 2w_1 abns + w_2 c^2 n^2 + w_2 d^2 s^2 + 2w_2 cdns$$

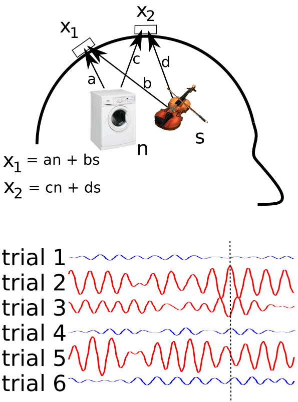
Why (and when) volume conduction matters



Case 2 (non-time-locked signals): Apply a linear classifier to some *non*-linear transformation of the raw data, for example power $\{x_1^2, x_2^2\}$:

$$
\begin{aligned}
f(\mathrm{X}) \quad = \quad & (w_1 a^2 + w_2 c^2) n^2 \\
& + (w_1 b^2 + w_2 d^2) s^2 \\
& + 2(w_1 ab + w_2 cd) ns
\end{aligned}
$$

Why (and when) volume conduction matters



Case 2 (non-time-locked signals): Apply a linear classifier to some *non*-linear transformation of the raw data, for example power $\{x_1^2, x_2^2\}$:

$$\begin{aligned} f(\mathrm{X}) \quad = \quad & (w_1 a^2 + w_2 c^2) n^2 \\ & + (w_1 b^2 + w_2 d^2) s^2 \\ & + 2(w_1 ab + w_2 cd) ns \end{aligned}$$

Some solutions (e.g. $w_1 = c^2, \quad w_2 = -a^2$) might cancel out the n^2 term; others (e.g. $w_1 = cd, \quad w_2 = -ab$) might cancel out the ns term, but we cannot remove both terms with any single solution.

Why (and when) volume conduction matters

When the features for classification consist of raw signal samples, or a linear transformation (detrending, bandpassing, ...), a linear classifier *might* be able to do your spatial filtering/source estimation for you.

Why (and when) volume conduction matters

When any non-linear transformation is to be used (amplitude, power, phase,...), then you must ensure linear spatial filtering is performed to estimate relevant sources before the non-linear step is applied.

Why (and when) volume conduction matters

When any non-linear transformation is to be used (amplitude, power, phase,. . .), then you must ensure linear spatial filtering is performed to estimate relevant sources before the non-linear step is applied.

...and gain 5–15 percentage-points in binary classification performance.

Why (and when) volume conduction matters

When any non-linear transformation is to be used (amplitude, power, phase,. . .), then you must ensure linear spatial filtering is performed to estimate relevant sources before the non-linear step is applied.

...and gain 5–15 percentage-points in binary classification performance.

- Static surface-Laplacian

Why (and when) volume conduction matters

When any non-linear transformation is to be used (amplitude, power, phase,. . .), then you must ensure linear spatial filtering is performed to estimate relevant sources before the non-linear step is applied.

...and gain 5–15 percentage-points in binary classification performance.

- Static surface-Laplacian
- Independent Components Analysis (ICA)

Why (and when) volume conduction matters

When any non-linear transformation is to be used (amplitude, power, phase,. . .), then you must ensure linear spatial filtering is performed to estimate relevant sources before the non-linear step is applied.

...and gain 5–15 percentage-points in binary classification performance.

- Static surface-Laplacian
- Independent Components Analysis (ICA)
- Beamforming to particular source locations

Why (and when) volume conduction matters

When any non-linear transformation is to be used (amplitude, power, phase,. . .), then you must ensure linear spatial filtering is performed to estimate relevant sources before the non-linear step is applied.

...and gain 5–15 percentage-points in binary classification performance.

- Static surface-Laplacian
- Independent Components Analysis (ICA)
- Beamforming to particular source locations
- Common Spatial Pattern (CSP)

Why (and when) volume conduction matters

When any non-linear transformation is to be used (amplitude, power, phase,. . .), then you must ensure linear spatial filtering is performed to estimate relevant sources before the non-linear step is applied.

...and gain 5–15 percentage-points in binary classification performance.

- Static surface-Laplacian
- Independent Components Analysis (ICA)
- Beamforming to particular source locations
- Common Spatial Pattern (CSP)
- All-in-one classifier approaches. . .

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. decide where the relevant source(s) are going to be;

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. decide where the relevant source(s) are going to be;
3. solve the inverse problem;

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. decide where the relevant source(s) are going to be;
3. solve the inverse problem;
4. extract features from the recovered sources;

Vapnik's Imperative

> When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.
>
> —Vladimir Vapnik

Should we...

1. measure the data;
2. decide where the relevant source(s) are going to be;
3. solve the inverse problem;
4. extract features from the recovered sources;
5. classify/regress against the desired output?

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. decide where the relevant source(s) are going to be;
3. solve the inverse problem;
4. extract features from the recovered sources;
5. classify/regress against the desired output?

Vapnik's Imperative

> When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.
>
> —Vladimir Vapnik

Should we...

1. measure the data;
2. decide where the relevant source(s) are going to be;
3. solve the inverse problem;
4. extract features from the recovered sources;
5. classify/regress against the desired output?

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. decide where the relevant source(s) are going to be;
3. "solve" the inverse problem;
4. extract features from the recovered sources;
5. classify/regress against the desired output?

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. estimate sources according to how good they are for predicting the given output;
3. extract features from the recovered sources;
4. classify/regress against the desired output.

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. estimate sources according to how good they are for predicting the given output;
3. classify/regress against the desired output, allowing the classification method to extract the most relevant features;

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. classify/regress against the desired output, allowing the classification method to extract the optimal source-estimation parameters as well as the most relevant features

Vapnik's Imperative

When solving a problem of interest, do not solve a harder/more general problem as an intermediate step.

—Vladimir Vapnik

Should we...

1. measure the data;
2. classify/regress against the desired output, allowing the classification method to extract the optimal source-estimation parameters as well as the most relevant features
3. solve the inverse problem if you still want to (to sanity-check/learn more about the result).

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathrm{tr}\left[\mathrm{S}\mathrm{S}^\top\right]$$

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathrm{tr}\left[\mathrm{D}\ \ \mathrm{S}\mathrm{S}^\top\right]$$

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathrm{tr}\left[\mathrm{D}\quad(\mathrm{W}\mathrm{X})(\mathrm{W}\mathrm{X})^\top\right]$$

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathsf{tr}\left[\mathrm{D} \quad \mathrm{W}\mathrm{X}\mathrm{X}^\top\mathrm{W}^\top\right]$$

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathsf{tr}\left[\mathrm{W}^\top \mathrm{D}\mathrm{W} \quad \mathrm{X}\mathrm{X}^\top\right]$$

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathsf{tr}\left[\mathrm{M} \quad \Sigma\right]$$

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \text{tr}\left[\mathrm{M} \quad \Sigma\right] = \mathrm{M}(:)^\top \Sigma(:)$$

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathsf{tr}\ [\mathrm{M}\ \ \Sigma] = \mathrm{M}(:)^\top \Sigma(:)$$

Use a good classifier to find M, then W has been found implicitly and can be recovered with an eigenvalue decomposition.

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathrm{tr}\left[\mathrm{W}^\top \mathrm{D}\mathrm{W} \quad \Sigma\right] = \mathrm{M}(:)^\top \Sigma(:)$$

Use a good classifier to find M, then W has been found implicitly and can be recovered with an eigenvalue decomposition.

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathsf{tr}\,[\mathrm{W}^{\top}\mathrm{D}\mathrm{W}\quad\Sigma] = \mathrm{M}(:)^{\top}\Sigma(:)$$

Use a good classifier to find M, then W has been found implicitly and can be recovered with an eigenvalue decomposition.

- Tomioka & Müller (2010), Neuroimage.
- Farquhar (2009), Neural Networks.

All-in-one approach for bandpower classifier

$$\mathrm{S} = \mathrm{W}\mathrm{X}$$

For linear classification of sources' bandpower, spatial filters can also be found automatically by a classifier:

$$f(\mathrm{X}) = \mathrm{tr}\,[\mathrm{W}^\top \mathrm{D}\mathrm{W} \;\; \Sigma] = \mathrm{M}(:)^\top \Sigma(:)$$

Use a good classifier to find M, then W has been found implicitly and can be recovered with an eigenvalue decomposition.

- Tomioka & Müller (2010), Neuroimage.
- Farquhar (2009), Neural Networks.

Similar formulations for other non-linear features??

Slightly deeper learning?

From Collobert & Weston's NIPS 2009 tutorial:

Engineering: complex features, simple algorithm.

vs

Machine-Learning: simple input, implicitly learn the features.

Slightly deeper learning?

From Collobert & Weston's NIPS 2009 tutorial:

Engineering: complex features, simple algorithm.

Preprocessing (spatial subspace, spectral filtering...) then *classification*

vs

Machine-Learning: simple input, implicitly learn the features.

Slightly deeper learning?

From Collobert & Weston's NIPS 2009 tutorial:

Engineering: complex features, simple algorithm.

Preprocessing (spatial subspace, spectral filtering...) then *classification*

vs

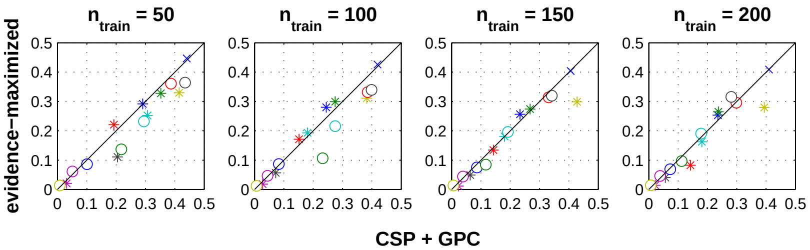
Machine-Learning: simple input, implicitly learn the features.

Idea: instead of performing CSP's least-square criterion to estimate discriminative sources

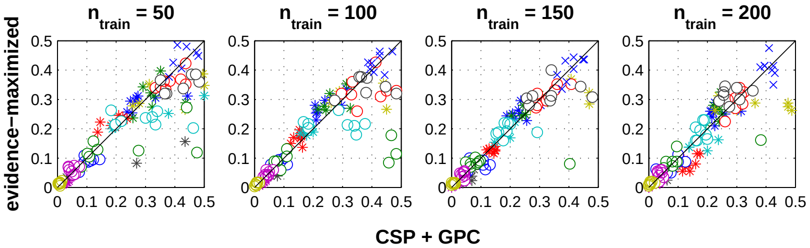
$$\mathrm{S} = \mathrm{WX}$$

then classifying the resulting bandpower features diag $(\mathrm{SS}^\top)$ according to some *other* loss function, let's treat W as the hyperparameters of (e.g.) a Gaussian Process classifier and optimize them according to the marginal-likelihood...

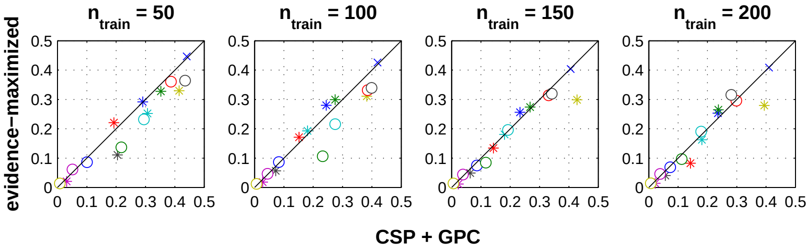
Slightly deeper learning?



Slightly deeper learning?



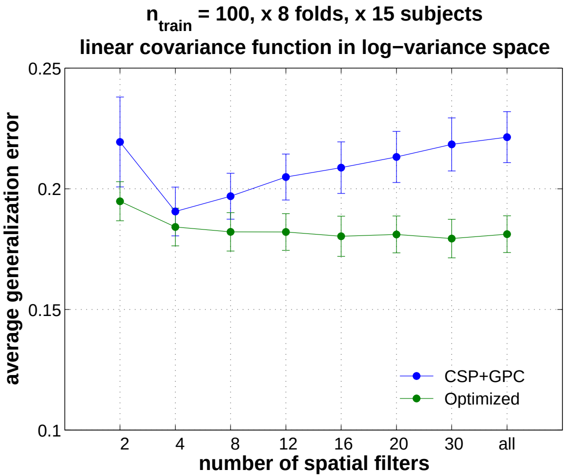
Slightly deeper learning?



Note:

- large individual variation
- particular benefits for smaller, noisier datasets.

Deeper learning $\rightsquigarrow$ more "hands-free" operation



Deeper still?

Automatic combination of/selection between first- and second-order features

- Christoforou et al. (2008) JMLR

Deeper still?

Automatic combination of/selection between first- and second-order features

- Christoforou et al. (2008) JMLR
- Tomioka & Müller (2010) Neuroimage

Deeper still?

Automatic combination of/selection between first- and second-order features

- Christoforou et al. (2008) JMLR
- Tomioka & Müller (2010) Neuroimage

Convex optimization of spatial filters, with automatic selection/weighting between frequency bands

- Tomioka & Müller (2010) Neuroimage

Deeper still?

Automatic combination of/selection between first- and second-order features

- Christoforou et al. (2008) JMLR
- Tomioka & Müller (2010) Neuroimage

Convex optimization of spatial filters, with automatic selection/weighting between frequency bands

- Tomioka & Müller (2010) Neuroimage
- Farquhar (2009) Neural Networks

Deeper still?

Automatic combination of/selection between first- and second-order features

- Christoforou et al. (2008) JMLR
- Tomioka & Müller (2010) Neuroimage

Convex optimization of spatial filters, with automatic selection/weighting between frequency bands

- Tomioka & Müller (2010) Neuroimage
- Farquhar (2009) Neural Networks
  - extensible to arbitrary number of dimensions (time, frequency, cross-subject, cross-condition, . . .)

Deeper still?

Automatic combination of/selection between first- and second-order features

- Christoforou et al. (2008) JMLR
- Tomioka & Müller (2010) Neuroimage

Convex optimization of spatial filters, with automatic selection/weighting between frequency bands

- Tomioka & Müller (2010) Neuroimage
- Farquhar (2009) Neural Networks
    - extensible to arbitrary number of dimensions (time, frequency, cross-subject, cross-condition, . . .)

Pre-processing can still make a difference to performance (e.g. equalizing variance across frequency bands to compensate for $1/f$; spatial pre-whitening in both first- and second-order cases).

Deeper still?

Automatic combination of/selection between first- and second-order features

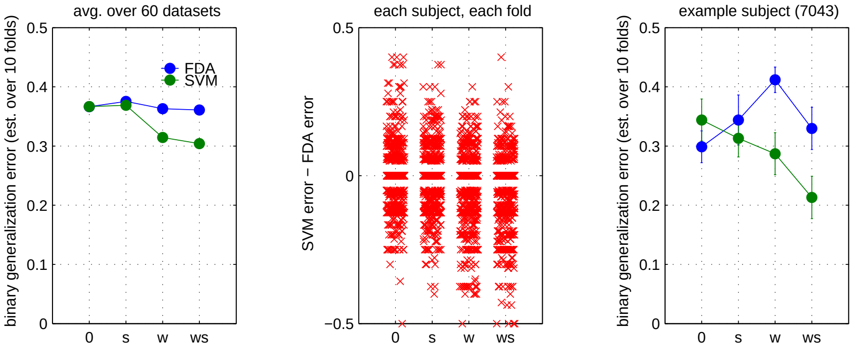
- Christoforou et al. (2008) JMLR
- Tomioka & Müller (2010) Neuroimage

Convex optimization of spatial filters, with automatic selection/weighting between frequency bands

- Tomioka & Müller (2010) Neuroimage
- Farquhar (2009) Neural Networks
  - extensible to arbitrary number of dimensions (time, frequency, cross-subject, cross-condition, ...)

Pre-processing can still make a difference to performance (e.g. equalizing variance across frequency bands to compensate for $1/f$; spatial pre-whitening in both first- and second-order cases).
Pre-processing the data can be seen as equivalent to changing the regularization environment. What is the "ideal" regularization strategy?

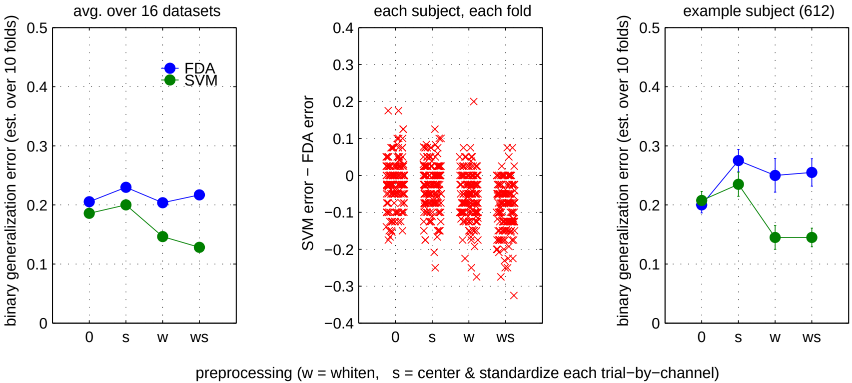
“Which classifier you use doesn’t matter”



preprocessing (w = whiten, s = center & standardize each trial−by−channel)

Felix Biessmann’s visual speller data (10 subjects x 6 stimulus conditions), offline analysis

“Which classifier you use doesn’t matter”



auditory ERP data, offline analysis

An Overfitting Nightmare?

- High noise

An Overfitting Nightmare?

- High noise
- Small number of data exemplars

An Overfitting Nightmare?

- High noise
- Small number of data exemplars
- Very large number of features.

An Overfitting Nightmare?

- High noise
- Small number of data exemplars
- Very large number of features.
  Well actually, the features are usually *highly* correlated.

An Overfitting Nightmare?

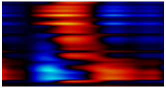
- High noise
- Small number of data exemplars
- Very large number of features.
  Well actually, the features are usually *highly* correlated.
  - This is a good thing—we only need to worry about a low-dimensional *subspace*.

An Overfitting Nightmare?

- High noise
- Small number of data exemplars
- Very large number of features.
  Well actually, the features are usually *highly* correlated.
  - This is a good thing—we only need to worry about a low-dimensional *subspace*.
  - This is a bad thing—can lead to trying to optimize very "stiff" systems.

Low-rank Classification

In linear ERP classification: classifier finds weights M for classifying space-$\times$-time "image" segments:

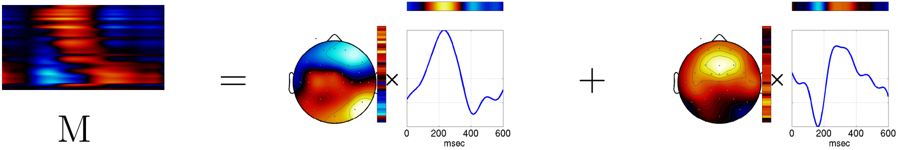


$$=$$

M

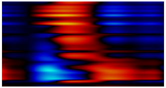
Low-rank Classification

In linear ERP classification: classifier finds weights M for classifying space-$\times$-time "image" segments:



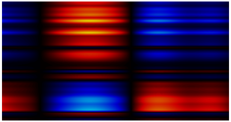
Low-rank Classification

In linear ERP classification: classifier finds weights M for classifying space-$\times$-time "image" segments:



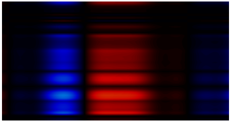
M

$=$



$\mathbf{w}_{s1}\ \mathbf{w}_{t1}^{\top}$

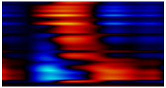
$+$



$\mathbf{w}_{s2}\ \mathbf{w}_{t2}^{\top}$

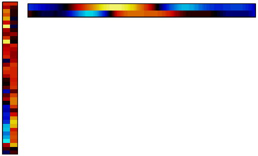
Low-rank Classification

In linear ERP classification: classifier finds weights M for classifying space-$\times$-time "image" segments:



M

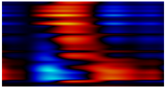
$=$



$\mathrm{W}_{\mathsf{s}} \; \mathrm{W}_{\mathsf{t}}^{\top}$

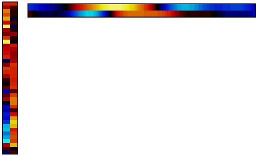
Low-rank Classification

In linear ERP classification: classifier finds weights M for classifying space-$\times$-time "image" segments:



M

$=$

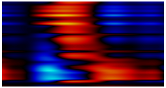


$\mathrm{W}_{\mathsf{s}} \; \mathrm{W}_{\mathsf{t}}^{\top}$

L_{Σ} regularization: regularize by putting an L-1 penalty on the singular values of M.

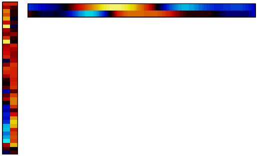
Low-rank Classification

In linear ERP classification: classifier finds weights M for classifying space-$\times$-time "image" segments:



M

$=$



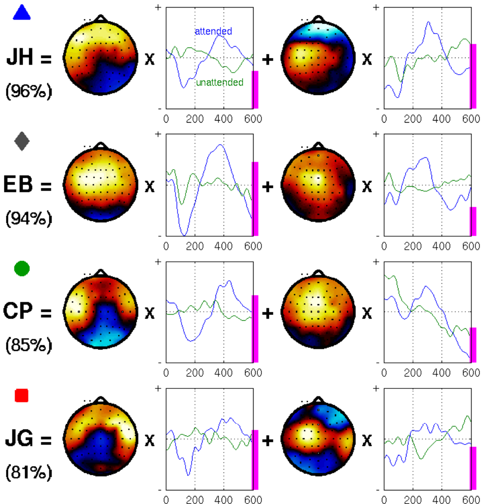
$\mathrm{W}_{\mathsf{s}}\ \mathrm{W}_{\mathsf{t}}^{\top}$

L_{Σ} regularization: regularize by putting an L-1 penalty on the singular values of M.

- Tomioka & Aihara (2007) ICML 2007.
- Tomioka & Müller (2010), Neuroimage.
- Farquhar (2009), Neural Networks.

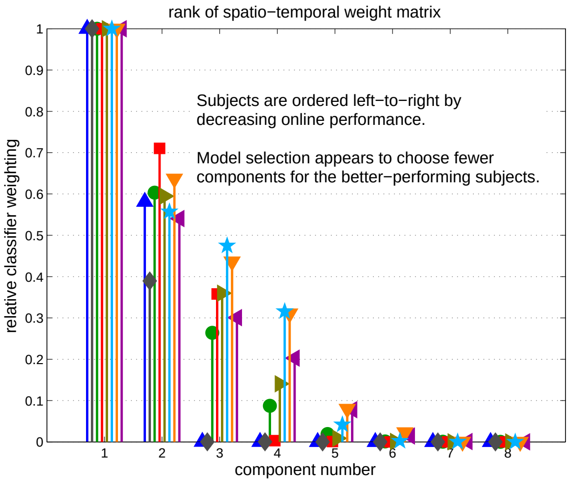
Example Sparsification Results

A BCI based on auditory stimuli (Hill et al., NIPS 2004 & BBCI Workshop 2009):



Example Sparsification Results

A BCI based on auditory stimuli (Hill et al., NIPS 2004 & BBCI Workshop 2009):



- In BCI, machine-learning methods allow us to optimize performance directly, avoiding the necessity to solve the inverse problem.

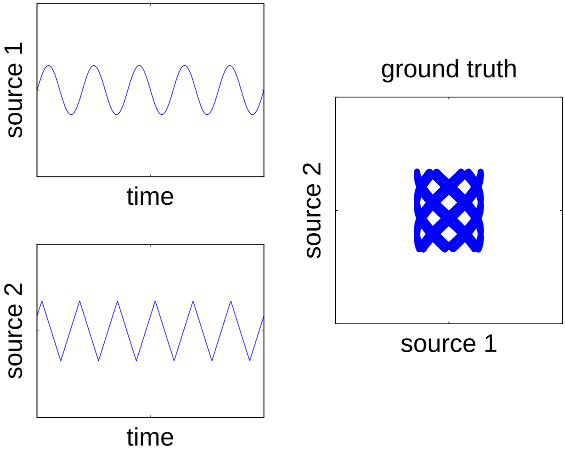
- In BCI, machine-learning methods allow us to optimize performance directly, avoiding the necessity to solve the inverse problem.
- Volume conduction must still be respected, especially when we use bandpower or other non-linear features.

- In BCI, machine-learning methods allow us to optimize performance directly, avoiding the necessity to solve the inverse problem.
- Volume conduction must still be respected, especially when we use bandpower or other non-linear features.
- Careful choice of classification methods can make a difference.

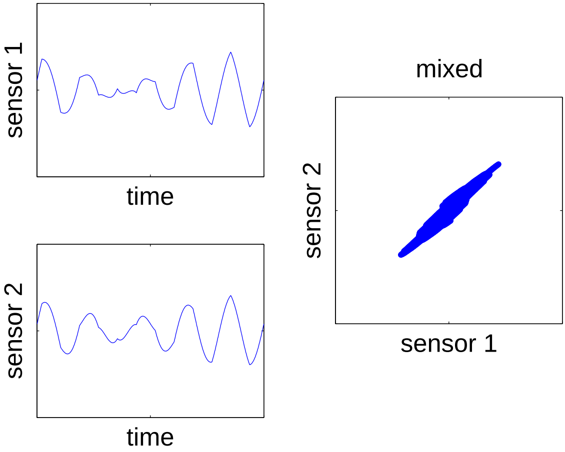
- In BCI, machine-learning methods allow us to optimize performance directly, avoiding the necessity to solve the inverse problem.
- Volume conduction must still be respected, especially when we use bandpower or other non-linear features.
- Careful choice of classification methods can make a difference.
- The better your classification method, the less you may need to worry about "preprocessing".

- In BCI, machine-learning methods allow us to optimize performance directly, avoiding the necessity to solve the inverse problem.
- Volume conduction must still be respected, especially when we use bandpower or other non-linear features.
- Careful choice of classification methods can make a difference.
- The better your classification method, the less you may need to worry about "preprocessing".
- Useful signals tend to live in low-dimensional subspaces, and optimizing directly for these can give an advantage in performance and in interpretability.

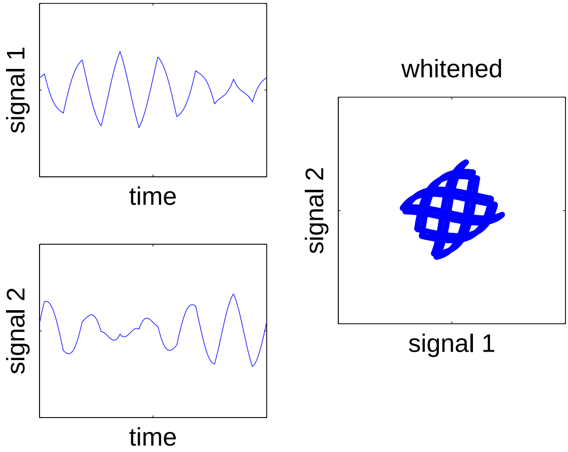
Source Separation



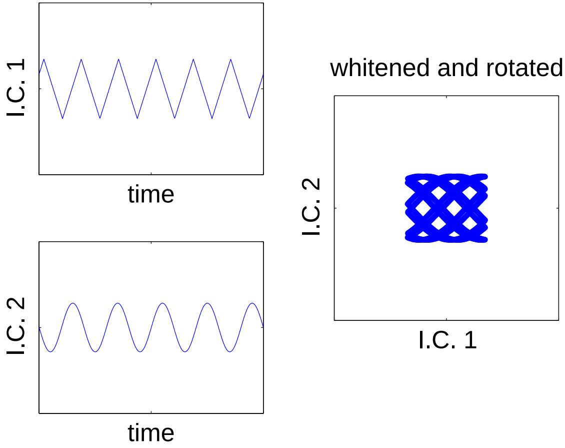
Source Separation



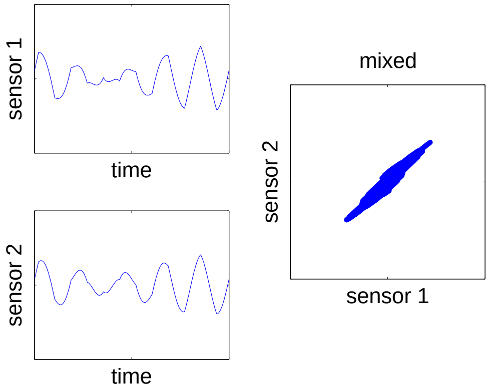
Source Separation



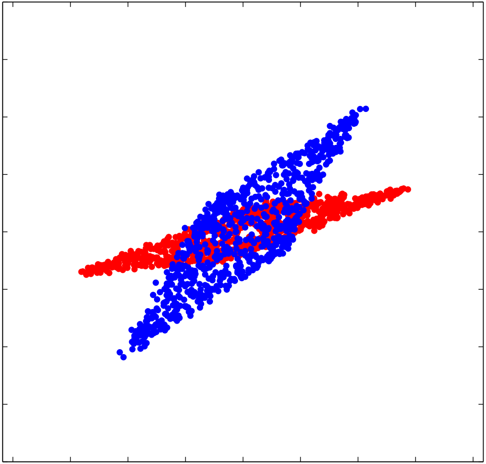
Source Separation



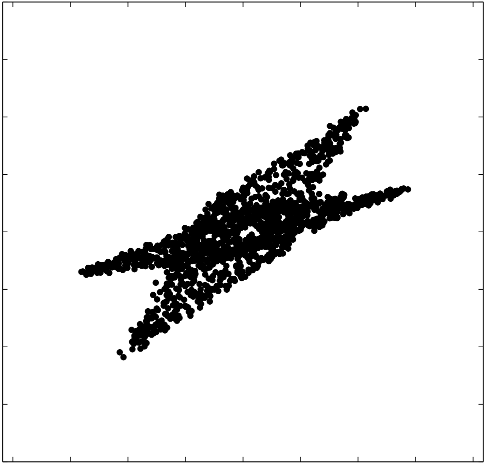
Cheap supervised rotation with CSP



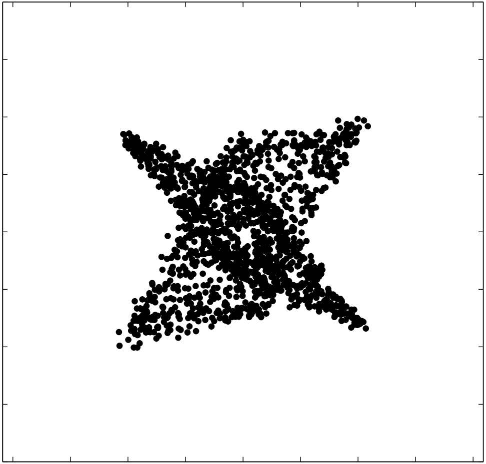
Cheap supervised rotation with CSP



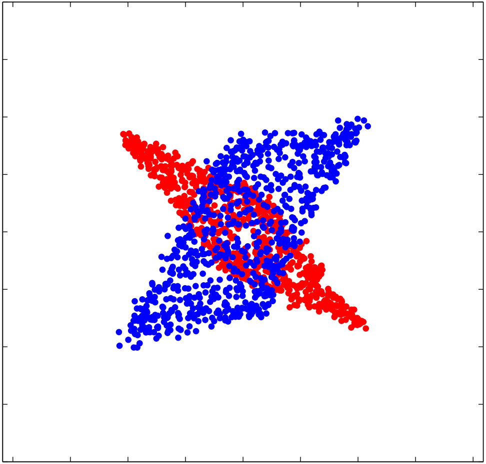
Cheap supervised rotation with CSP



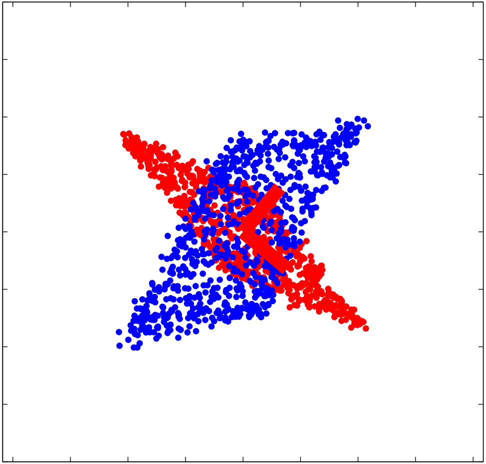
Cheap supervised rotation with CSP



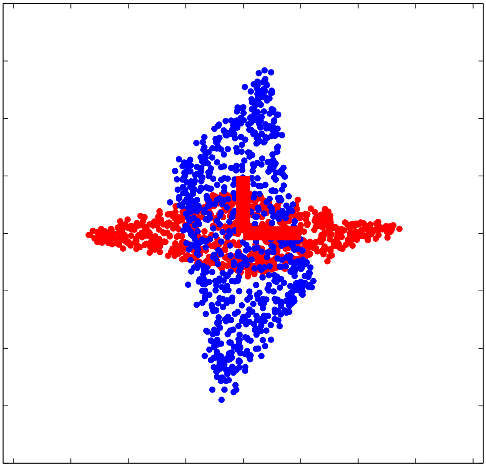
Cheap supervised rotation with CSP



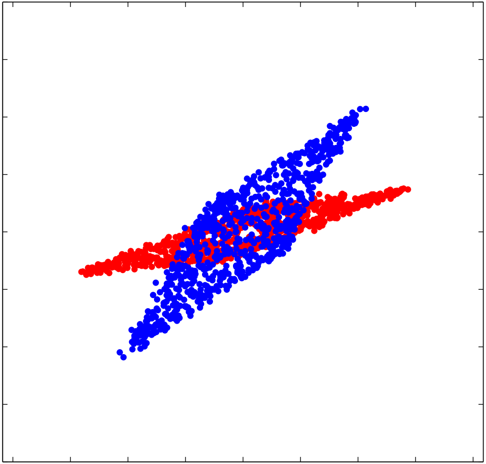
Cheap supervised rotation with CSP



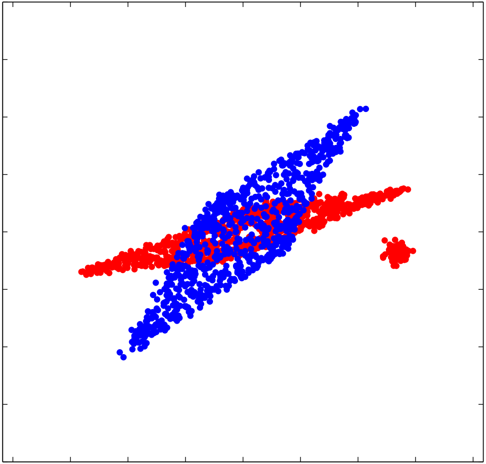
Cheap supervised rotation with CSP



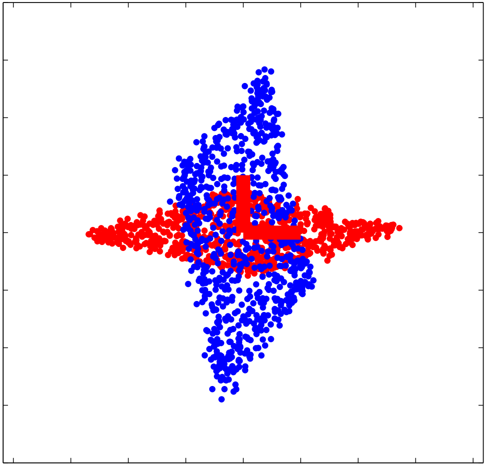
CSP: outlier- (artifact-) sensitivity



CSP: outlier- (artifact-) sensitivity



CSP: outlier- (artifact-) sensitivity



CSP: outlier- (artifact-) sensitivity