Donders Institute
for Brain, Cognition and Behaviour

Brain Reading (MKI43)

Lecture 3: Advanced discriminative approaches

Marcel van Gerven
Assistant Professor
Distributed Representations Group
Donders Centre for Cognition

Radboud University Nijmegen

Discriminant classifiers:

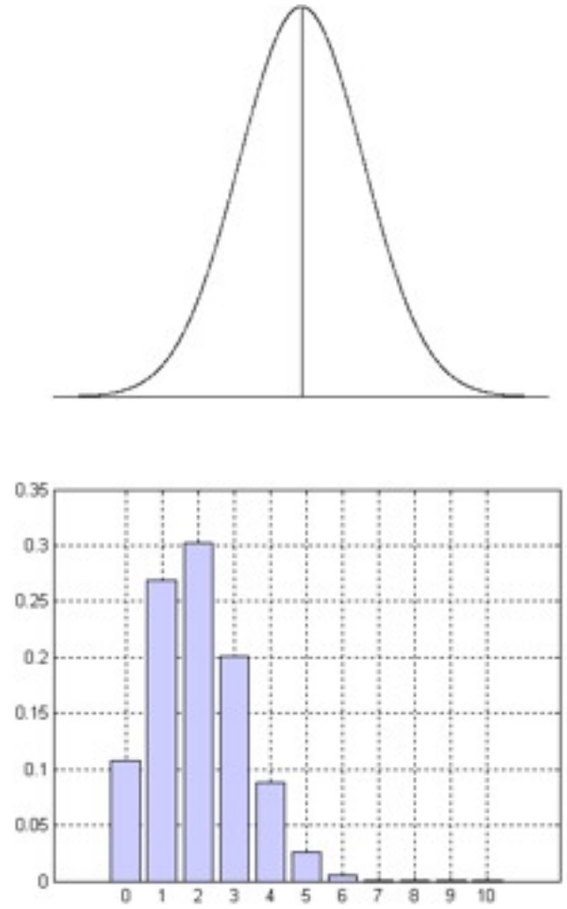
- extensions often nontrivial
- no probabilistic outputs
- blackbox method

predictive density

condition data

$$p(x \mid \boldsymbol{y}, D)$$

measurements



select most likely value

Computing the predictive density: MAP approach

$$p(x \mid \boldsymbol{y}, D) \approx p(x \mid \boldsymbol{y}, \boldsymbol{\theta}^*)$$

where $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid D)$

is the maximum a posteriori (MAP) estimate

Example: Binary logistic regression

Predicted class:

$$\hat{x} = \arg\max_x p(x \mid \boldsymbol{y}, \boldsymbol{\theta})$$

with $x \in \{0, 1\}$

Decision boundary:

$$p(X = 0 \mid \boldsymbol{y}, \boldsymbol{\theta}) = p(X = 1 \mid \boldsymbol{y}, \boldsymbol{\theta})$$

Equivalent to:

$$\frac{p(X = 1 \mid \boldsymbol{y}, \boldsymbol{\theta})}{p(X = 0 \mid \boldsymbol{y}, \boldsymbol{\theta})} = 1$$

Example: Logistic regression

Assume a linear decision boundary:

$$\log \frac{p(X=1 \mid \boldsymbol{y}, \boldsymbol{\theta})}{p(X=0 \mid \boldsymbol{y}, \boldsymbol{\theta})} = \theta_1 y_1 + \cdots + \theta_p y_p$$

log odds ratio:

positive values point to class 1 and negative values point to class 0

Note: $\alpha = \text{logit}(p) = \log \frac{p}{1-p}$

$$p = \text{logit}^{-1}(\alpha) = \frac{1}{1+\exp(-\alpha)}$$

Hence

$$\begin{aligned} p(X=1 \mid \boldsymbol{y}, \boldsymbol{\theta}) &= \frac{1}{1+\exp(-\theta_1 y_1 - \cdots - \theta_p y_p)} \\ &= (1+\exp(-\boldsymbol{\theta}^T \boldsymbol{y}))^{-1} \end{aligned}$$

with $x \in \{0, 1\}$. Often, a constant term is part of $\boldsymbol{y}$

Example: Logistic regression

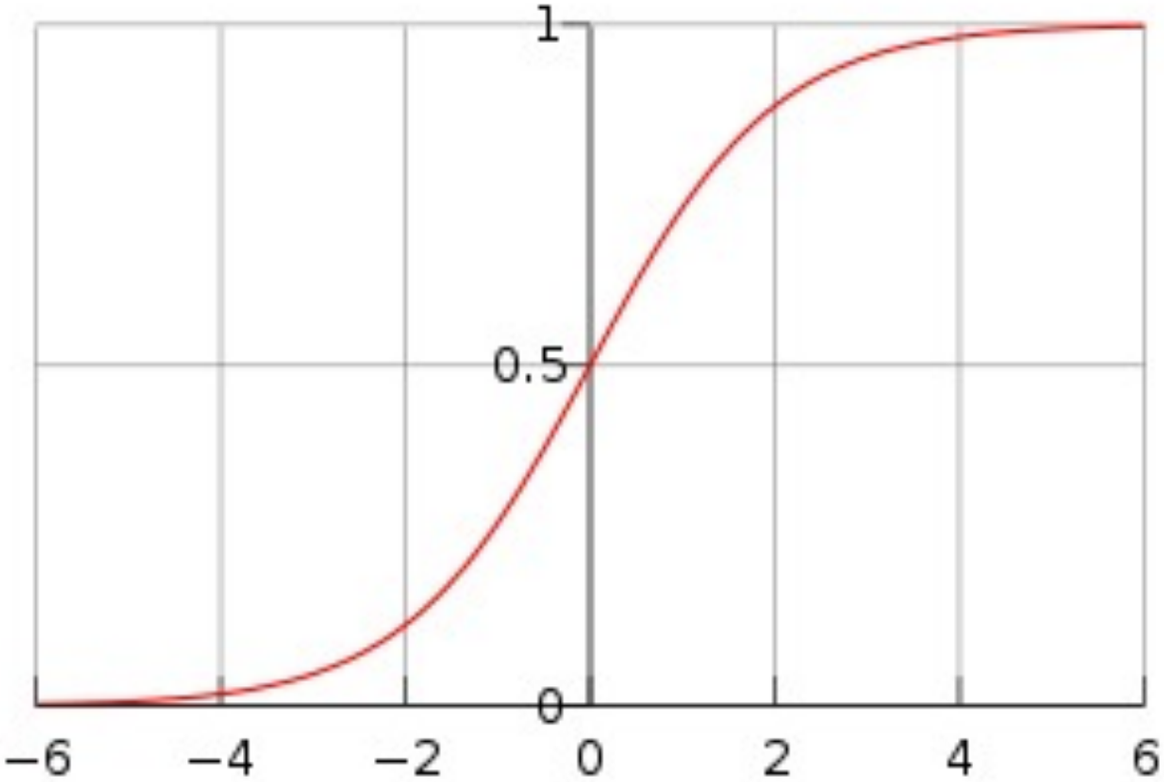


Figure 1. The logistic function, with *z* on the horizontal axis and *f(z)* on the vertical axis

MAP parameter estimation

model parameters ↑ $\boldsymbol{\theta}$

likelihood ↑ $p(D \mid \boldsymbol{\theta})$

prior ↑ $p(\boldsymbol{\theta})$

$$p(\boldsymbol{\theta} \mid D) = \frac{p(D \mid \boldsymbol{\theta})p(\boldsymbol{\theta})}{p(D)}$$

$p(D)$ → evidence

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \boldsymbol{D}) \\
&= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{D} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta}) \\
&= \arg\max_{\boldsymbol{\theta}} \log \{p(\boldsymbol{D} \mid \boldsymbol{\theta}) P(\boldsymbol{\theta})\} \\
&= \arg\max_{\boldsymbol{\theta}} \{\log p(\boldsymbol{D} \mid \boldsymbol{\theta}) + \log P(\boldsymbol{\theta})\}
\end{aligned}$$

Define:

$$J(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + R(\boldsymbol{\theta})$$

where the right-hand side represent the log probability terms

Parameter estimation for the MAP is equivalent to maximization of $J(\boldsymbol{\theta})$

$$L(\boldsymbol{\theta}) = \log p(D \mid \boldsymbol{\theta})$$

$$= \sum_{n=1}^{N} \log p(x^{(n)}, \boldsymbol{y}^{(n)} \mid \boldsymbol{\theta})$$

assumption: $p(D \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} p(x^{(n)}, \boldsymbol{y}^{(n)} \mid \boldsymbol{\theta})$

For discriminative models we always observe $\boldsymbol{y}^{(n)}$ so instead we can use the **conditional** log likelihood:

$$L(\theta) = \sum_{n=1}^{N} \log p(x^{(n)} \mid \boldsymbol{y}^{(n)}, \boldsymbol{\theta})$$

$$R(\boldsymbol{\theta}) = \log p(\boldsymbol{\theta})$$

The prior allows us to express constraints on the classifier parameters. For now we assume a flat prior:

$$p(\boldsymbol{\theta}) \propto 1$$

In that case,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

the **maximum likelihood estimate** of the parameters

Example: MLE for logistic regression

Note:

$$p(X = 1 \mid \boldsymbol{y}, \boldsymbol{\theta}) = (1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{y}))^{-1}$$
$$p(X = 0 \mid \boldsymbol{y}, \boldsymbol{\theta}) = 1 - p(X = 1 \mid \boldsymbol{y}, \boldsymbol{\theta}) = (1 + \exp(\boldsymbol{\theta}^T \boldsymbol{y}))^{-1}$$

$$\log p(X = 0 \mid \boldsymbol{y}, \boldsymbol{\theta}) = -\log(1 + \exp(\boldsymbol{\theta}^T \boldsymbol{y}))$$
$$\log p(X = 1 \mid \boldsymbol{y}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \boldsymbol{y} - \log(1 + \exp(\boldsymbol{\theta}^T \boldsymbol{y}))$$

MLE

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

where

$$L(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log p(x^{(n)} \mid \boldsymbol{y}^{(n)}, \boldsymbol{\theta})$$
$$= \sum_{n=1}^{N} \left\{ x^{(n)} \log p(X^{(n)} = 1 \mid \boldsymbol{y}^{(n)}, \boldsymbol{\theta}) + (1 - x^{(n)}) \log p(X^{(n)} = 0 \mid \boldsymbol{y}^{(n)}, \boldsymbol{\theta}) \right.$$
$$= \sum_{n=1}^{N} x^n \boldsymbol{\theta}^T \boldsymbol{y}^{(n)} - \log(1 + \exp(\boldsymbol{\theta}^T \boldsymbol{y}^{(n)}))$$

Example: MLE for logistic regression

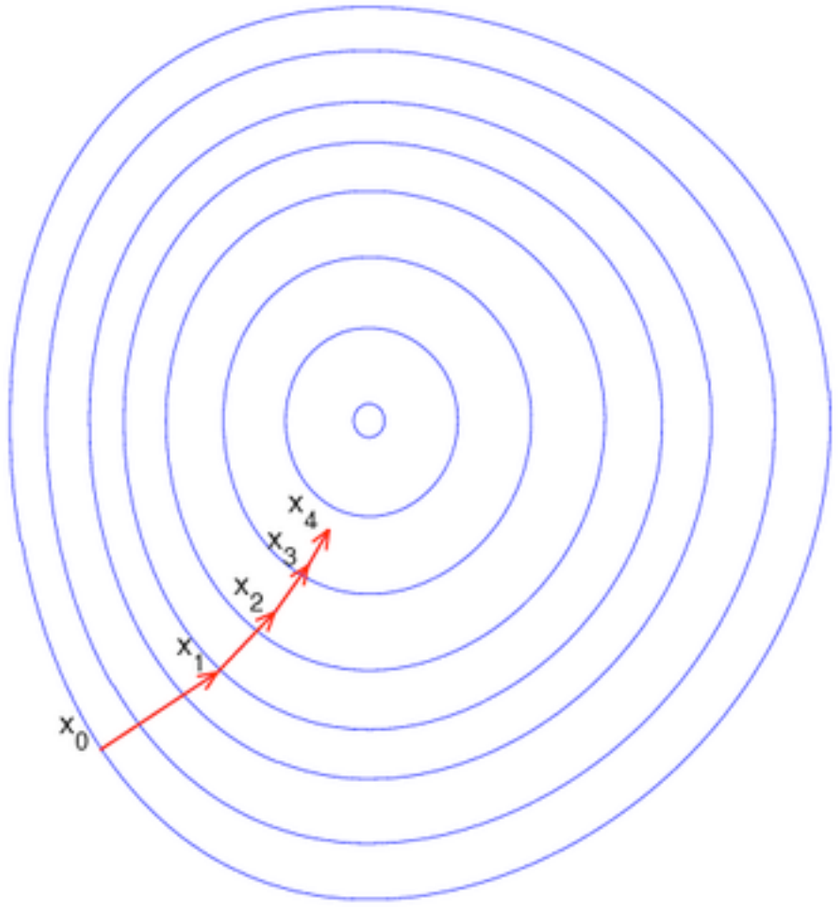
So

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left\{ \sum_{n=1}^{N} x^n \boldsymbol{\theta}^T \boldsymbol{y}^{(n)} - \log(1 + \exp(\boldsymbol{\theta}^T \boldsymbol{y}^{(n)})) \right\}$$

How do we find this maximum?

Gradient ascent:

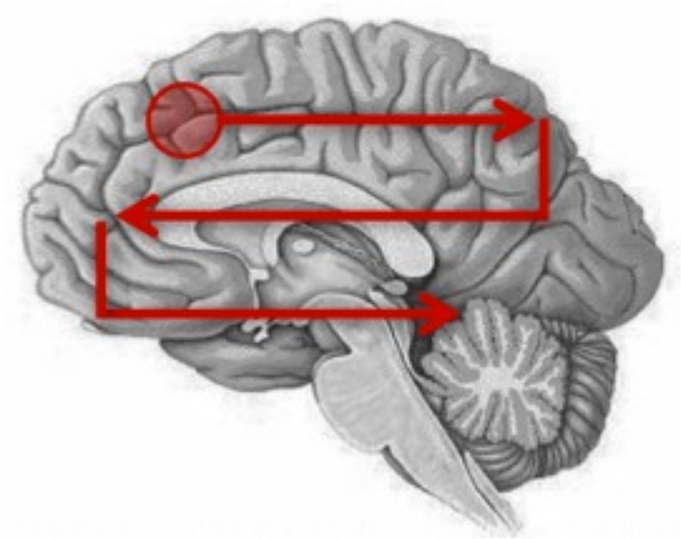
$$\theta_i^{(t+1)} \leftarrow \theta_i^{(t)} + \eta \frac{\partial L(\boldsymbol{\theta})}{\partial \theta_i^{(t)}}$$



Out of the scope of this course...

How to use the logistic regression model?

Searchlight



- Estimate MLE parameters for each sphere
- Compute predictions for each sphere
- Visualize the results

But...

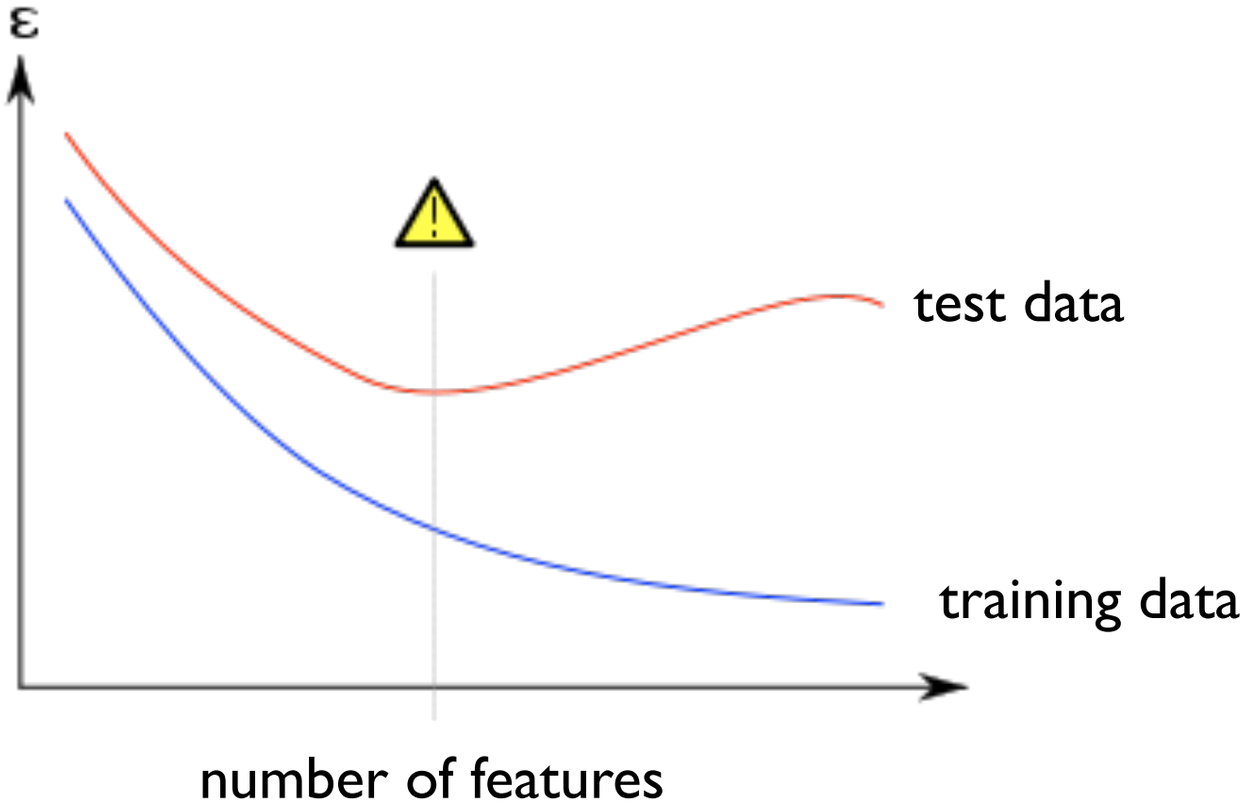
- Local estimates can give suboptimal performance
- We might miss truly distributed responses

How to use the logistic regression model?

Whole-brain approach

Could we use all the voxels as input to logistic regression?

Nope... overfitting...



Possible solution: feature selection

Formally, selection of a subset $F' \subseteq F$ of the features $F = \{x_1, \ldots, x_n\}$.

Reduction of the dimensionality may increase generalization performance.

Example: wrapper methods

Algorithm 2 Wrapper (Backward Elimination), *returns a feature set S of size N.*

Input: feature set Ω
Set $S = \Omega$
repeat
  Remove each feature $X \in S$, one-by-one, and evaluate your learning algorithm with the reduced set S.
  Identify the feature that allows the smallest drop in performance.
  Discard that feature.
until $|S| = N$

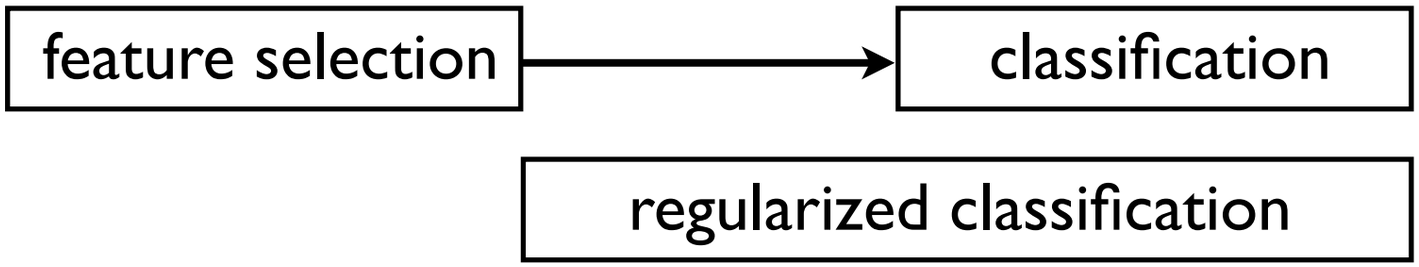
E.g. recursive feature elimination:

De Martino F, Valente G, Staeren N, Ashburner J, Goebel R., Formisano E. (2008). Combining multivariate voxel selection and support vector machines for mapping and classification of fMRI spatial patterns. *NeuroImage*, **43**, 44-48.

Another solution: regularization

Regularization: force classifier parameters to be well-behaved

Does not require two separate analysis steps



Recall MAP estimation:

$$\begin{aligned} \log p(\boldsymbol{\theta} \mid \boldsymbol{D}) &\propto \log p(\boldsymbol{D} \mid \boldsymbol{\theta}) + \log P(\boldsymbol{\theta}) \\ &= L(\boldsymbol{\theta}) + R(\boldsymbol{\theta}) \end{aligned}$$

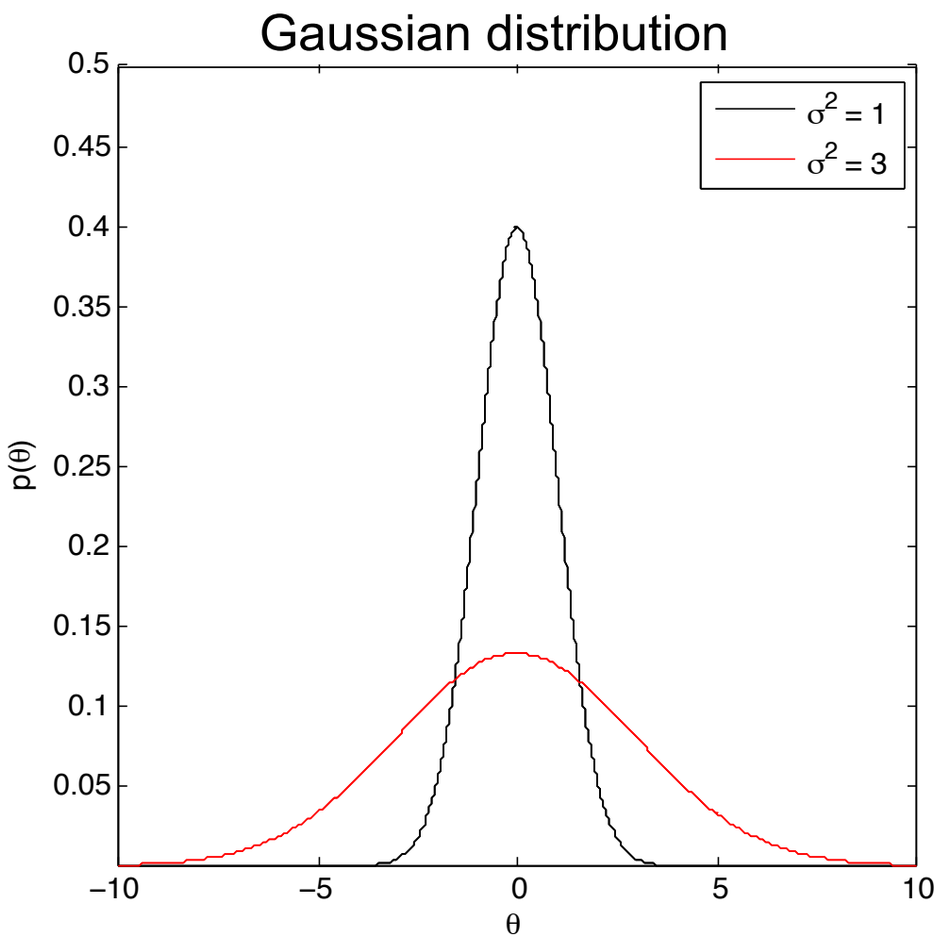
Regularization can be achieved by expressing constraints on $P(\boldsymbol{\theta})$

E.g. *theta should have small values* or *theta should have many zeros*

Small values

For each parameter assume:

$$P(\theta \mid \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\theta^2}{2\sigma^2}\right)$$



What does $R(\boldsymbol{\theta})$ look like?

Small values

$$
\begin{aligned}
R(\boldsymbol{\theta}) &= \log P(\boldsymbol{\theta}) \\
&= \log \prod_i P(\theta_i) \\
&= \sum_i \log \left\{ \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\theta_i^2}{2\sigma^2} \right) \right\} \\
&\propto -\frac{1}{2\sigma^2} \sum_i \theta_i^2 \\
&\propto -\frac{1}{2\sigma^2} ||\boldsymbol{\theta}||_2^2
\end{aligned}
$$

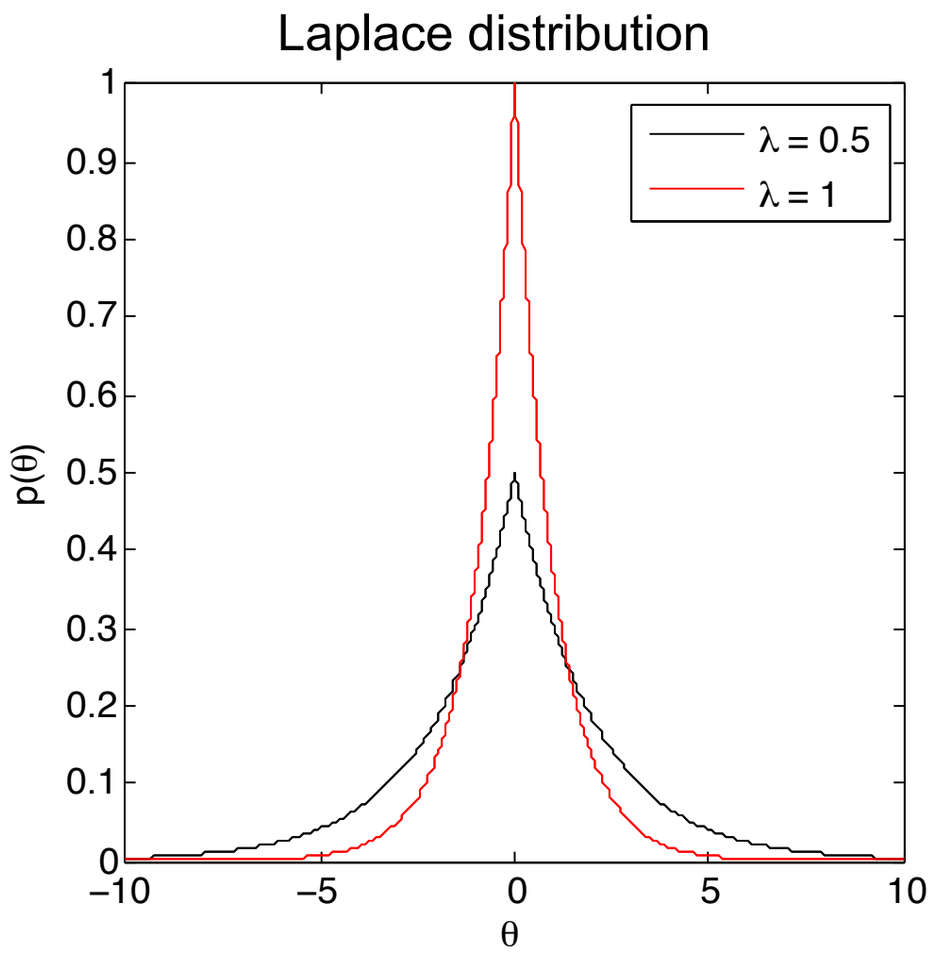
with $||\boldsymbol{x}||_p = (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$

Hence:

$$
\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \left\{ L(\boldsymbol{\theta}) - \lambda ||\boldsymbol{\theta}||_2^2 \right\}
$$

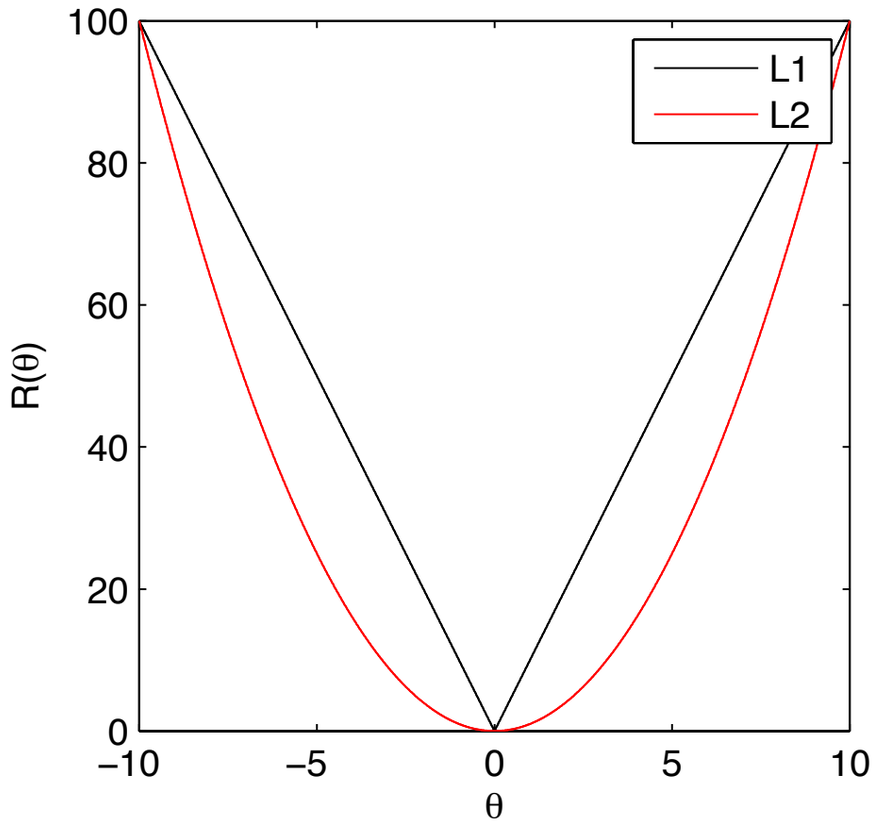
For each parameter assume:

$$P(\theta \mid \lambda) = \lambda \exp\left(-\lambda|\theta|\right)$$

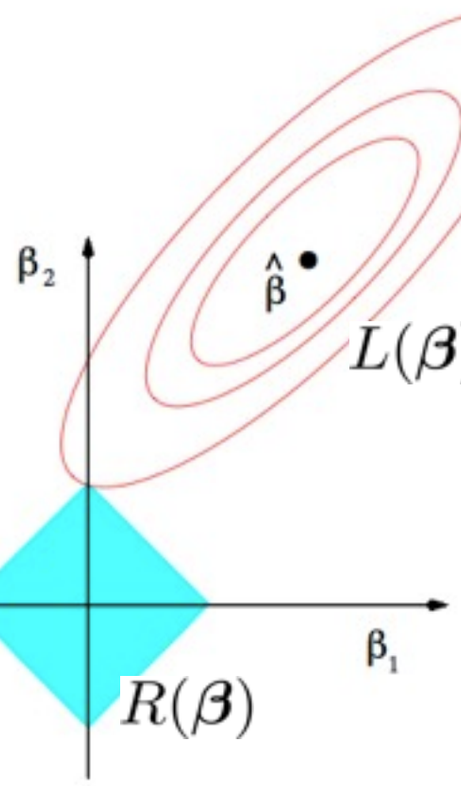


$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \{L(\boldsymbol{\theta}) - \lambda||\boldsymbol{\theta}||_1\}$$

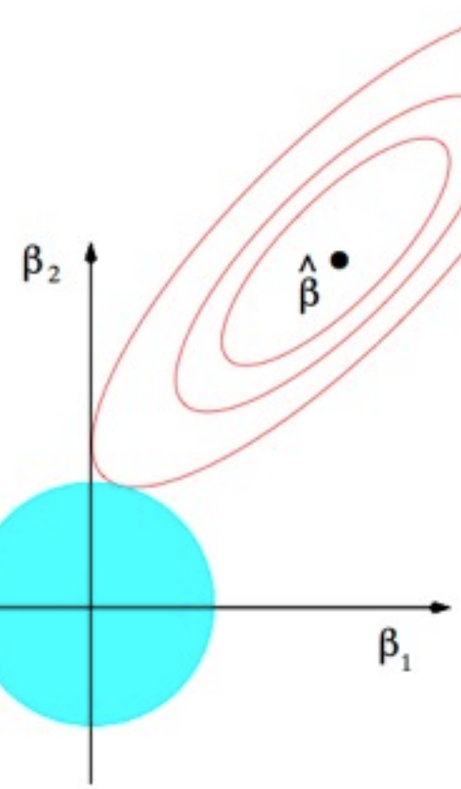
L1 versus L2 regularization



L1 regularization



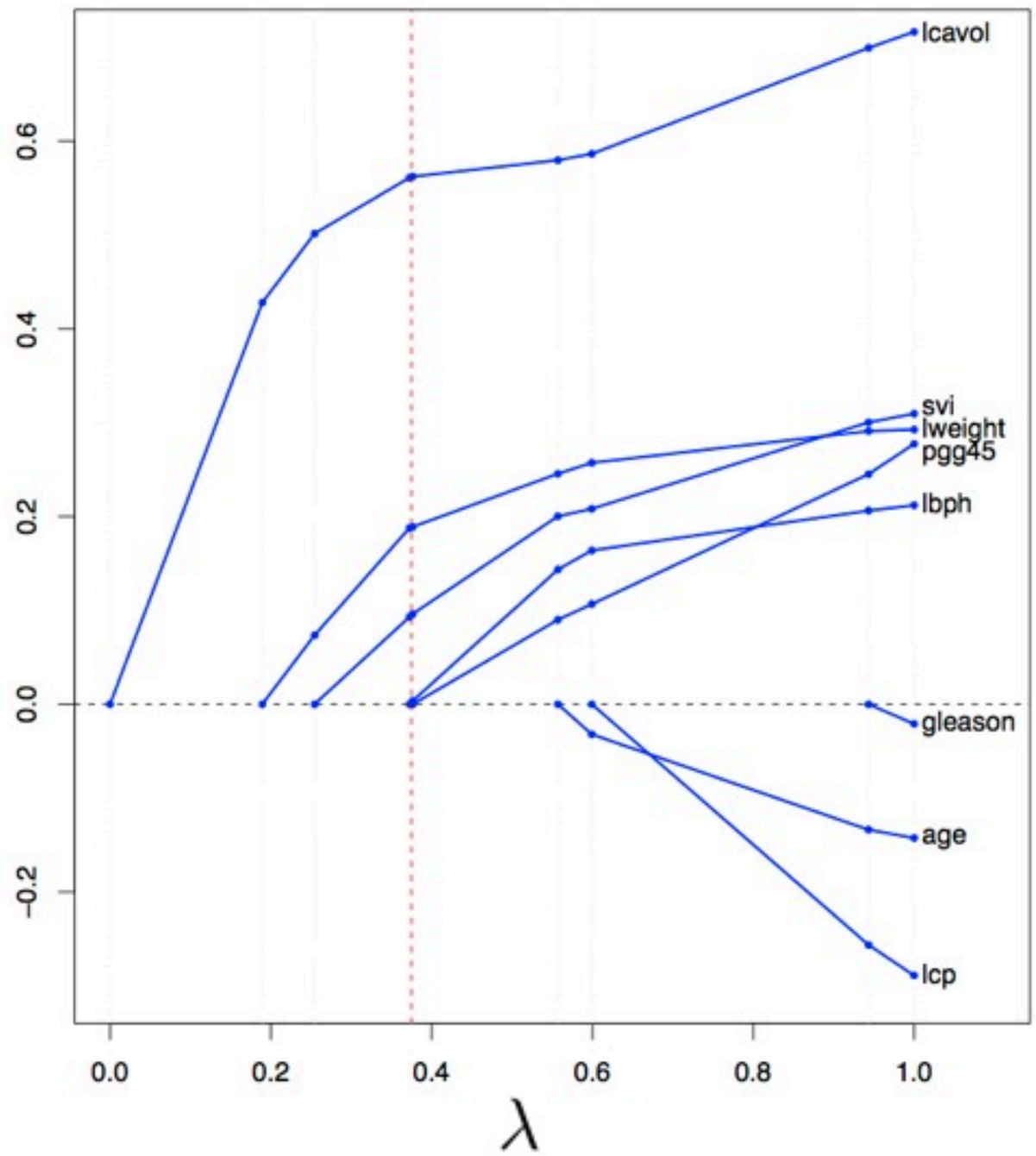
L2 regularization



- L1 regularization performs feature selection
- L2 regularization is robust in the presence of correlated variables

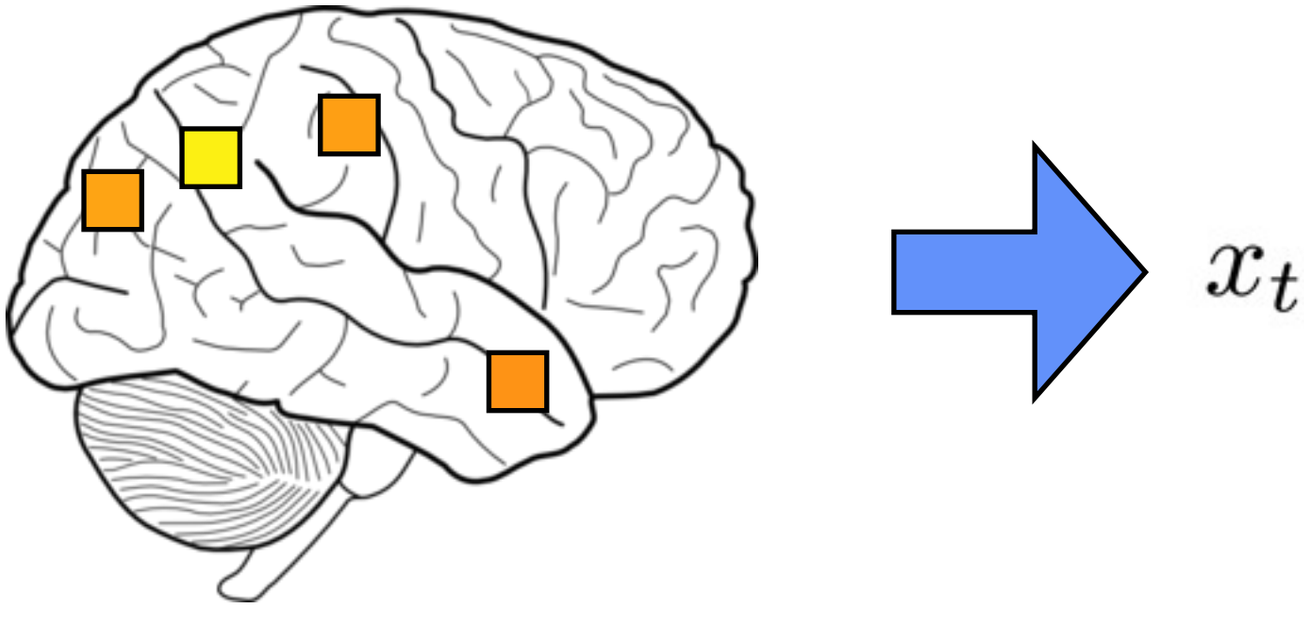
The L1 regularization path

θ^*



How to choose λ ?

Summary: regularization approach



θ

Posterior on the parameters:

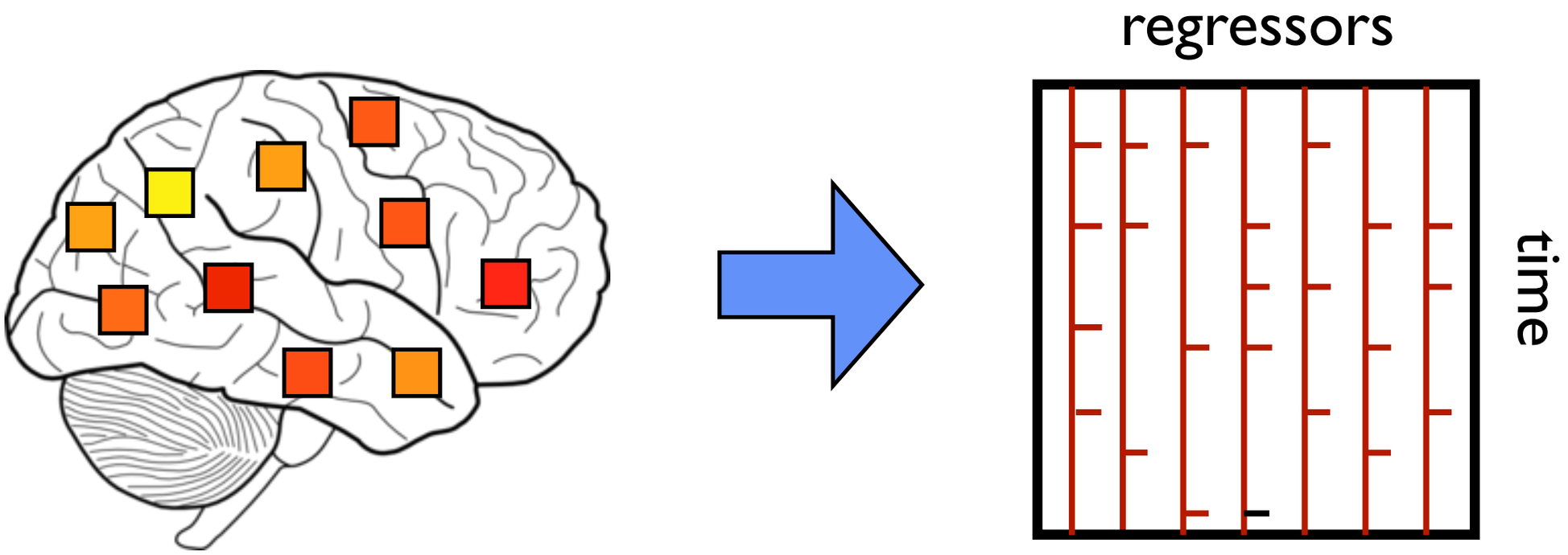
$$P(\theta \mid D) \propto P(D \mid \theta)p(\theta)$$

Sparseness and/or smoothness constraints on θ via $p(\theta)$:

- MAP regularization (ridge, lasso, elastic net, graphnet, total variation,...)
- Bayesian variable selection (spike-and-slab, ARD, MVL prior,..)

From classification to reconstruction

Predict from a very high-dimensional input (fMRI voxels) to a (possibly) very high-dimensional output (image pixels).

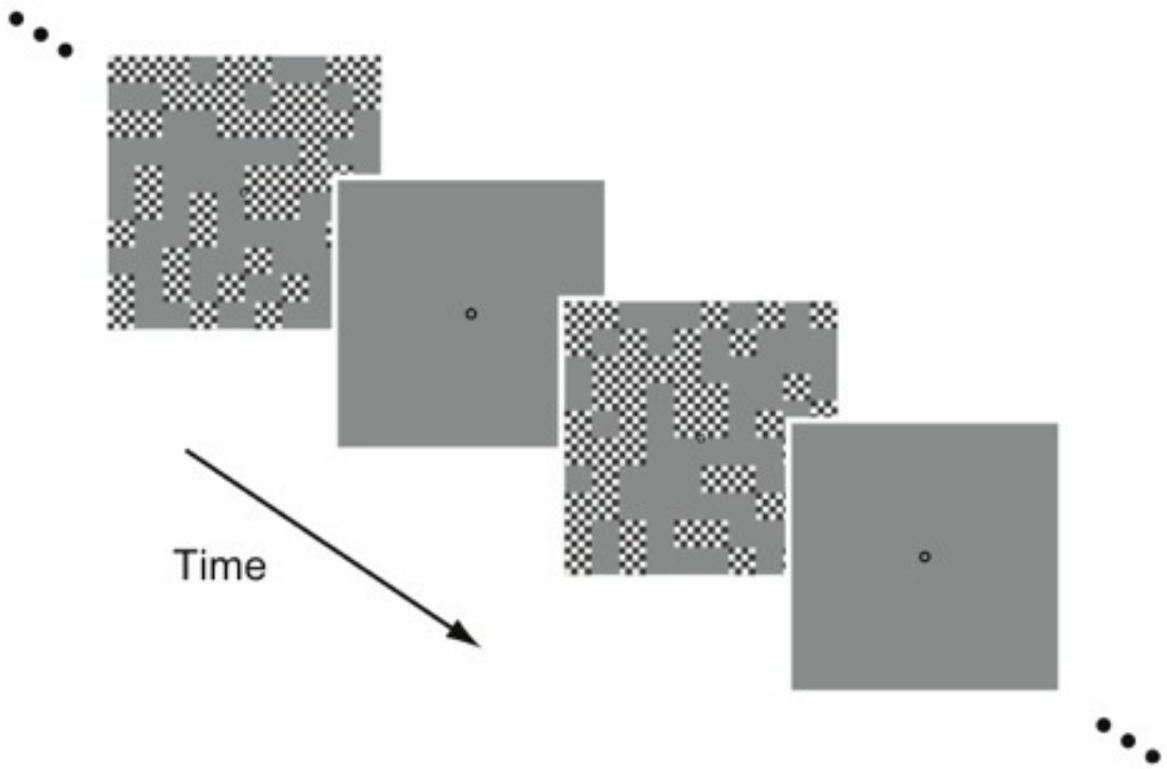


$$x^* = \arg\max_x p(x \mid y)$$

Discriminative approach

Discriminative approach: multiscale image decoding

training data



test data

Y. Miyawaki, H. Uchida, O. Yamashita, M. Sato, Y. Morito, H. C. Tanabe, N. Sadato, and Y. Kamitani. Visual image reconstruction from human brain activity using a combination of multiscale local image decoders. *Neuron*, 60(5):915–929, 2008.

How to solve the reconstruction problem?

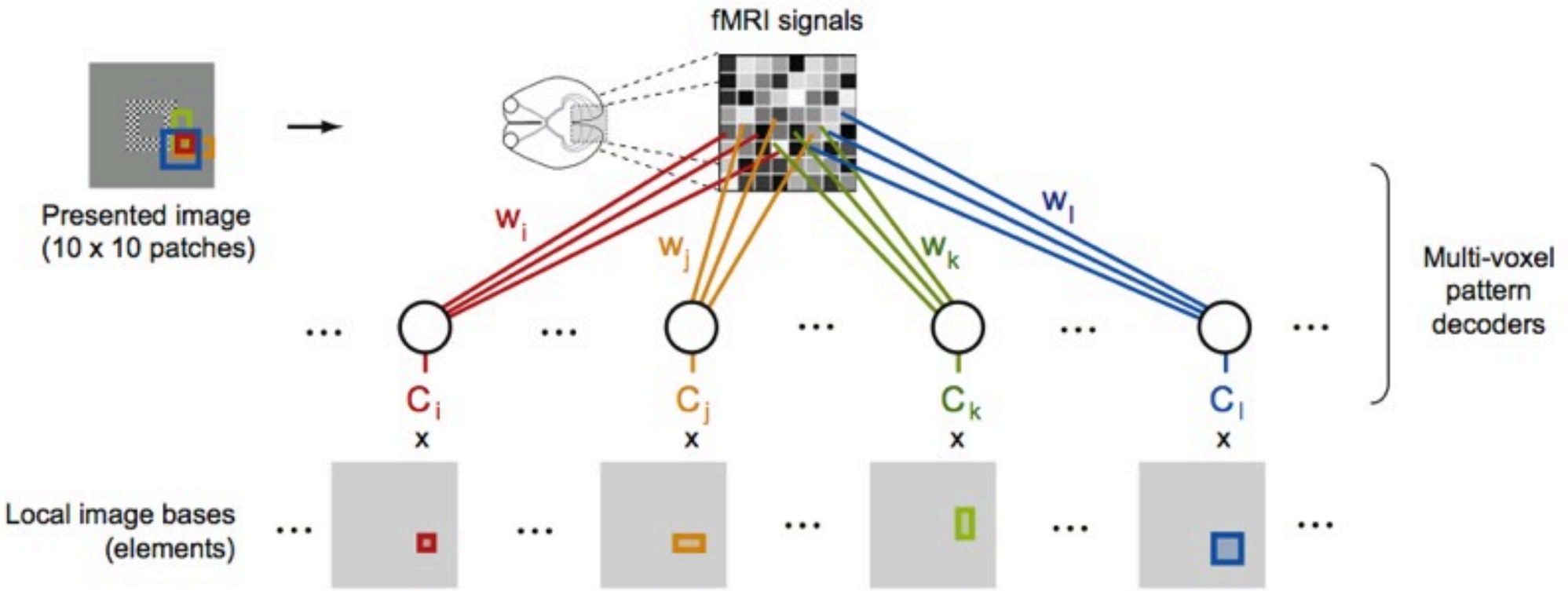
Presented image
(10 x 10 patches)

How to solve the reconstruction problem?

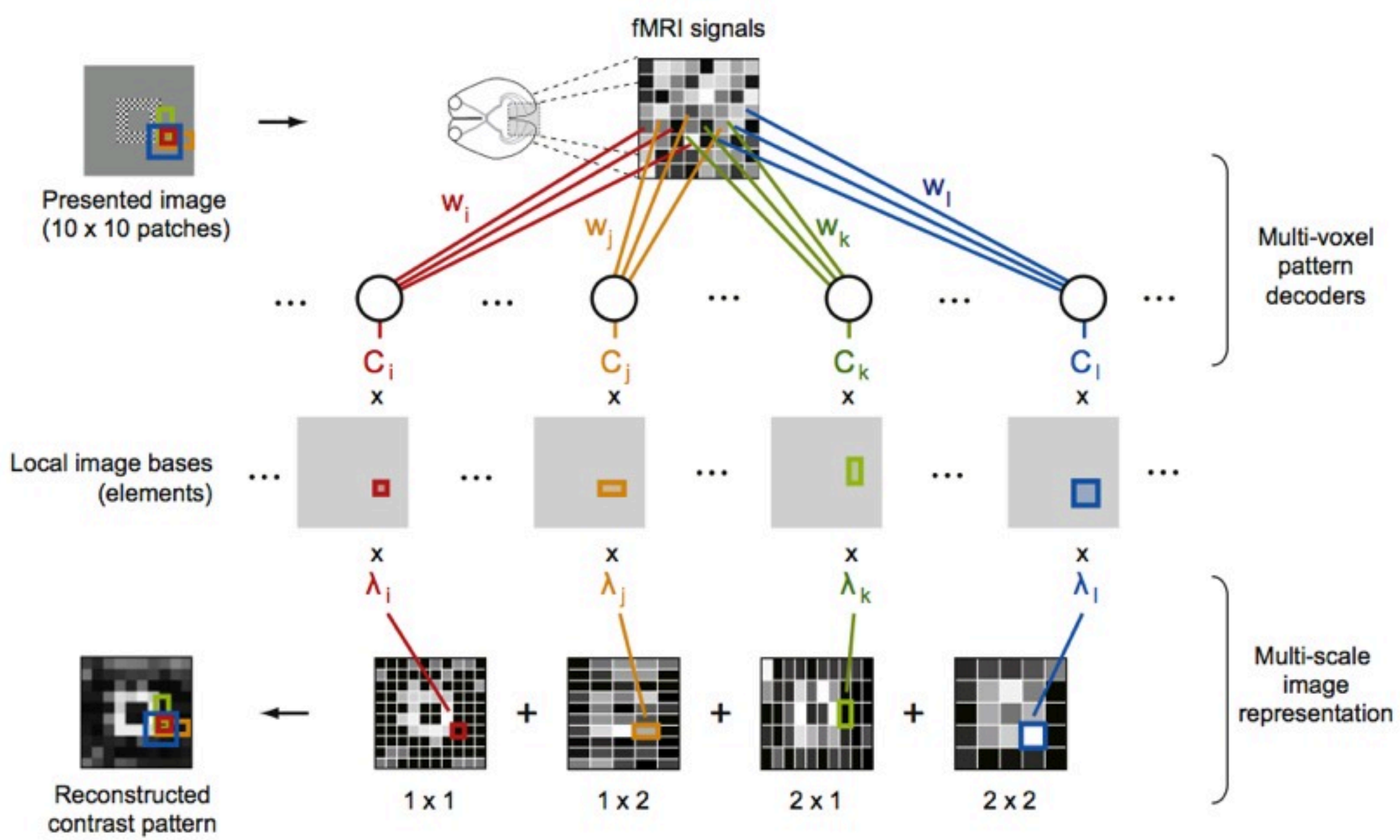
Presented image
(10 x 10 patches)

Local image bases
(elements)

How to solve the reconstruction problem?



How to solve the reconstruction problem?



Mathematical details

The mean contrast value of each local image element was defined as the number of flickering patches divided by the total number of patches (1x1,[0 or 1];1x2 and 2x1,[0,0.5, or 1];2x2,[0,0.25,0.5,0.75, or 1]).

probability of a contrast:

$$p_{\mathbf{w}}(k|\mathbf{r}) = \frac{\exp\left[y_{\mathbf{w}_k}(\mathbf{r})\right]}{\sum_{j}^{K} \exp\left[y_{\mathbf{w}_j}(\mathbf{r})\right]} \qquad \text{where} \quad y_{\mathbf{w}_k}(\mathbf{r}) = \sum_{d}^{D} w_k^d r^d + w_k^0$$

Conventional approach; maximize likelihood.

Combination coefficients were determined by the least squares method using a training data set.

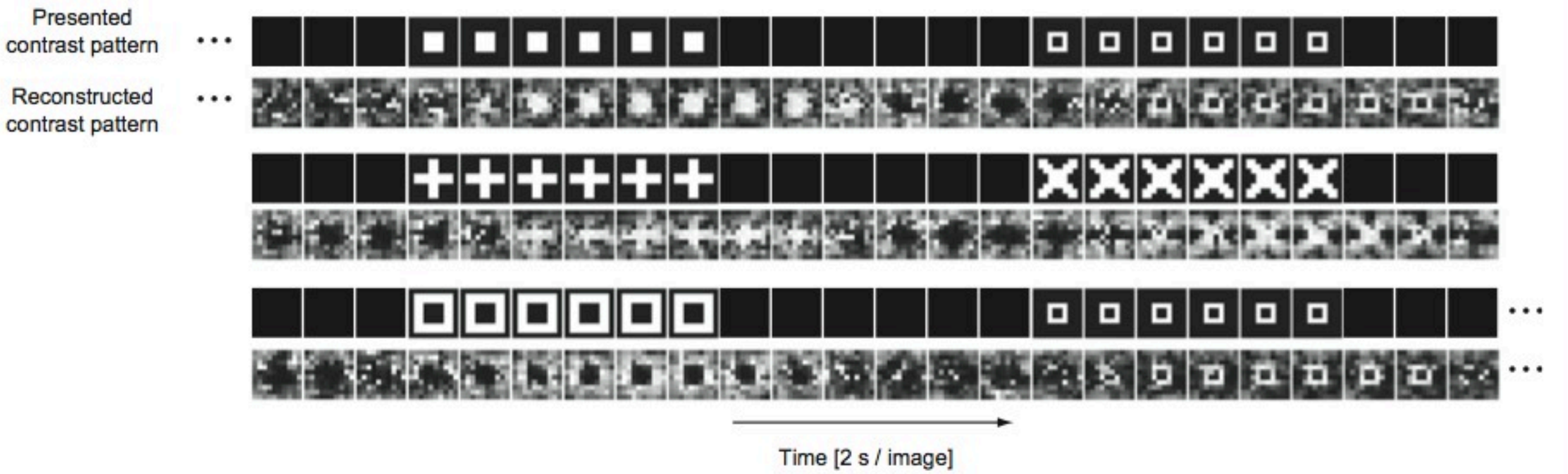
$$\widehat{I}(x|\mathbf{r}) = \sum_{m} \lambda_m C_m(\mathbf{r}) \phi_m(x)$$

combination coefficient

predicted contrast for patch m

is position x part of the patch m

Results


Presented contrast pattern
Reconstructed contrast pattern
Time [2 s / image]

Results

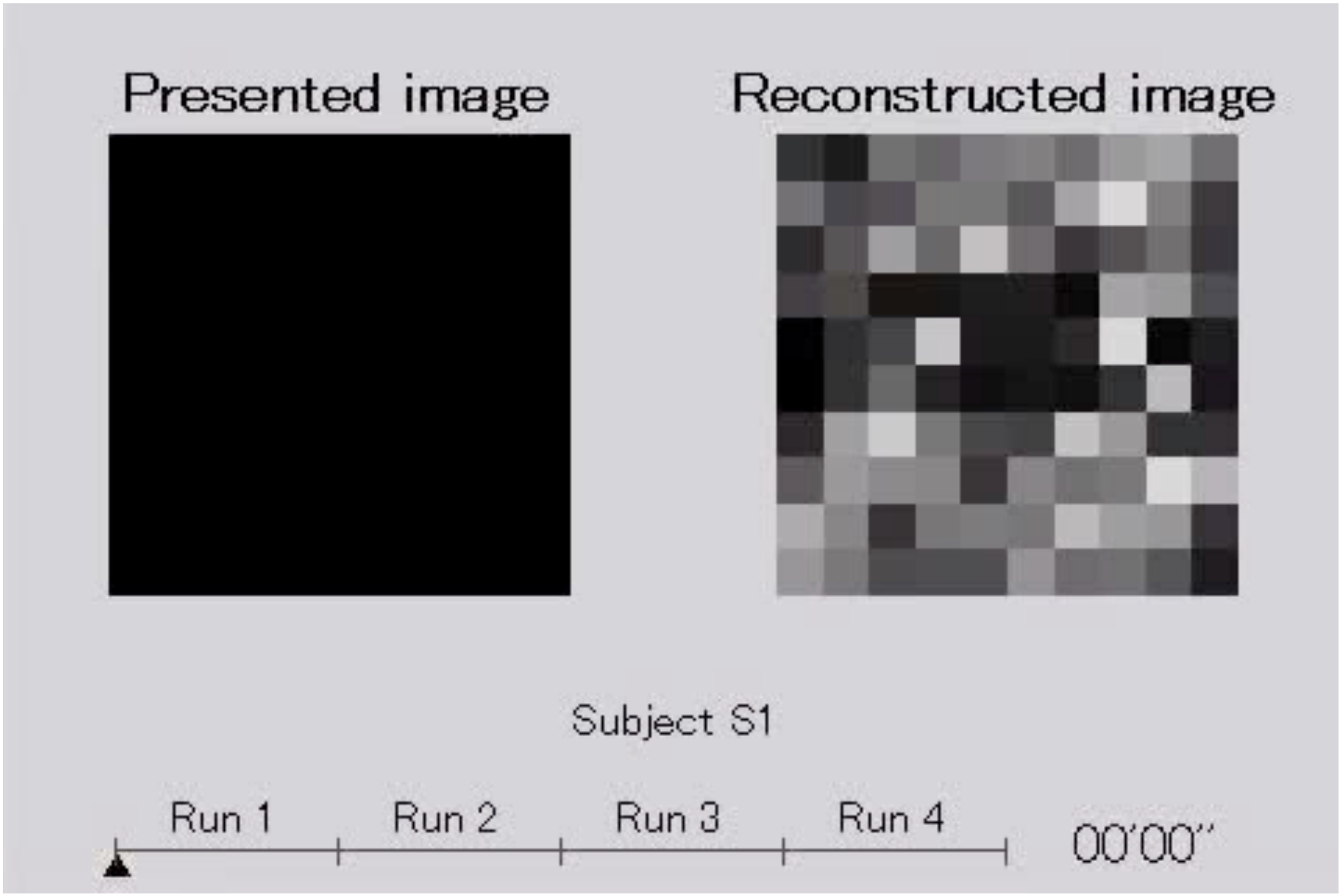
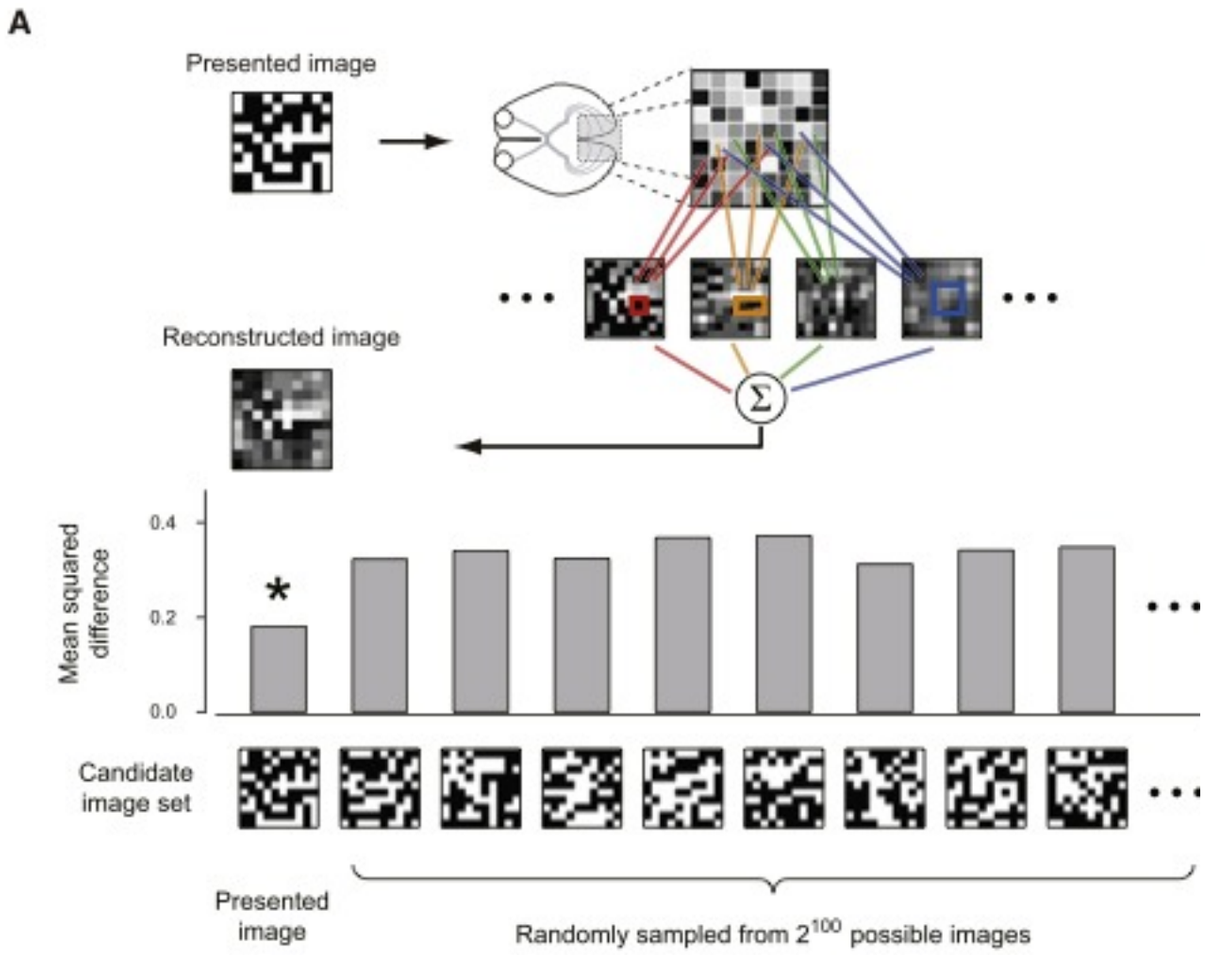
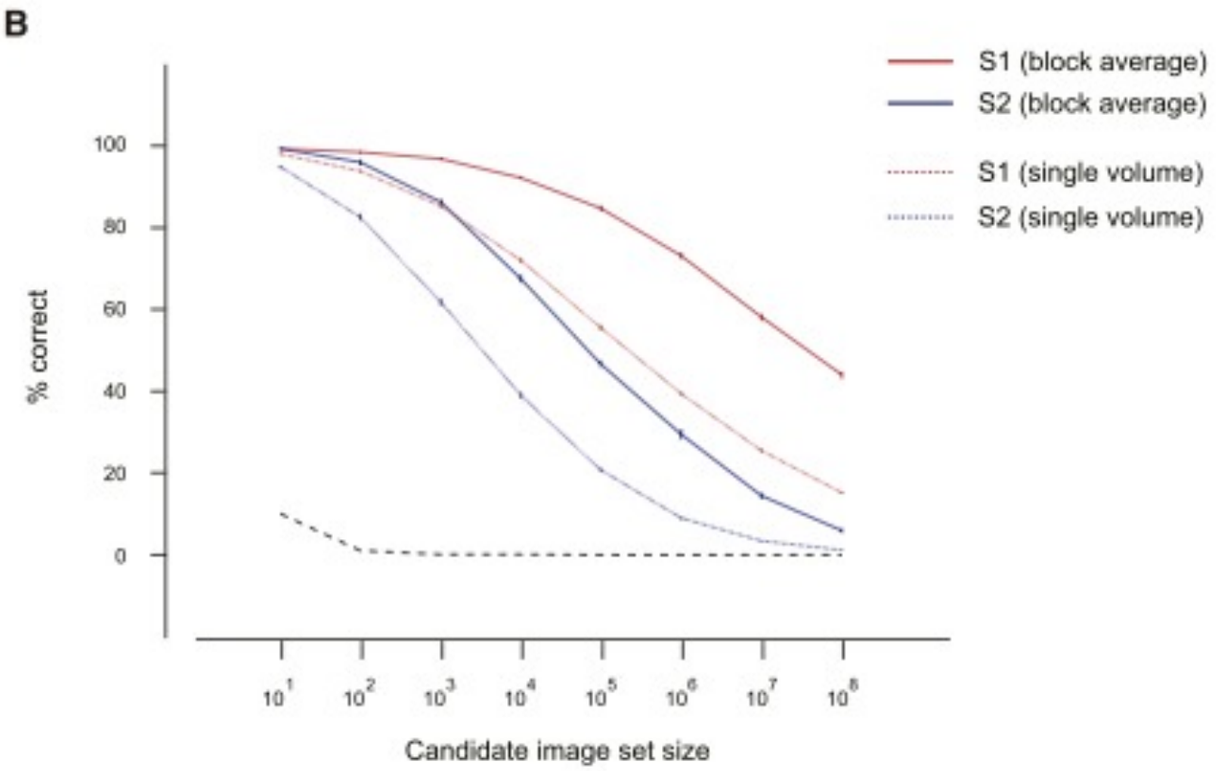
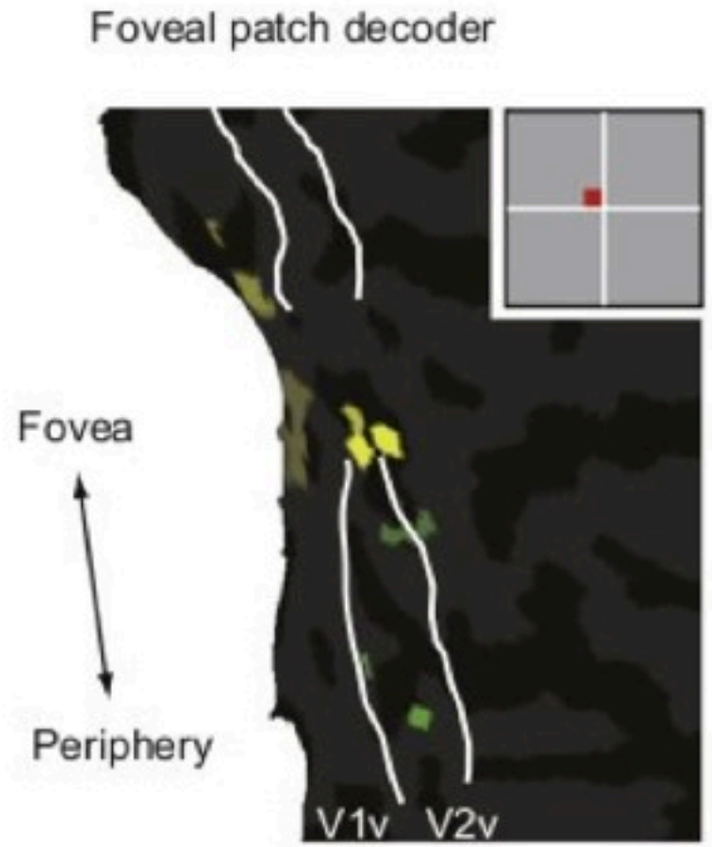

Presented image
Reconstructed image
Subject S1
Run 1
Run 2
Run 3
Run 4
00'00''

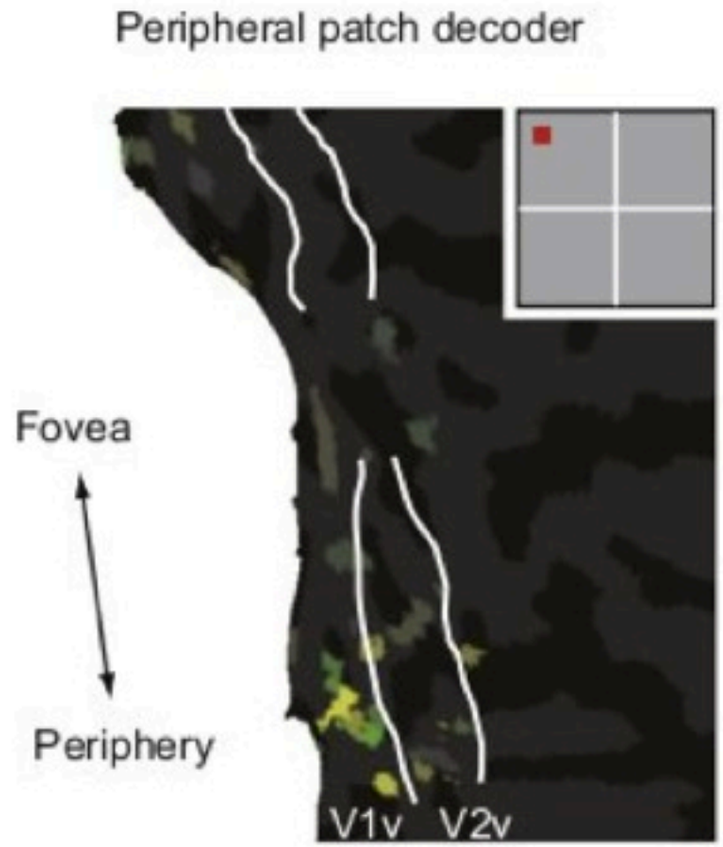
Image identification

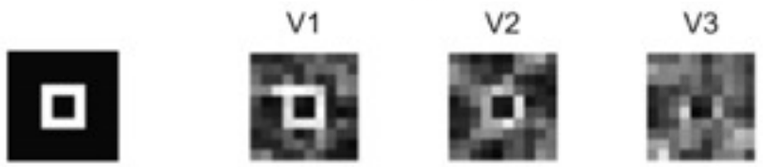


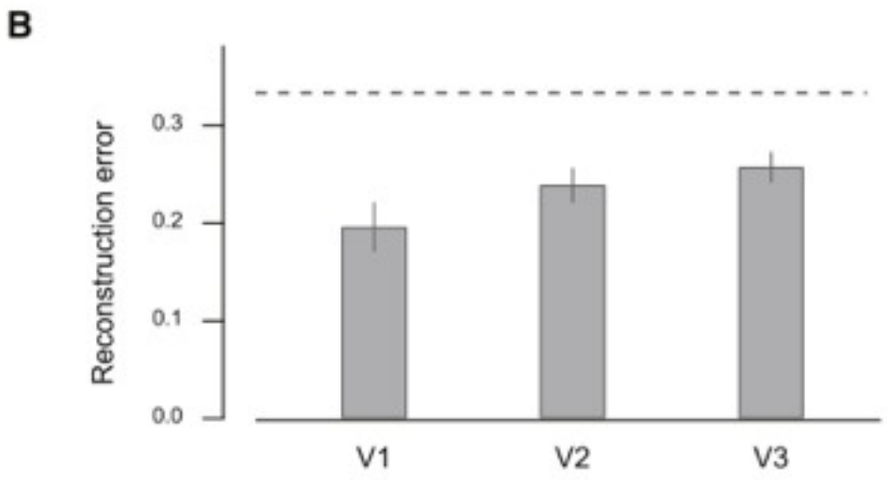


Regional involvement

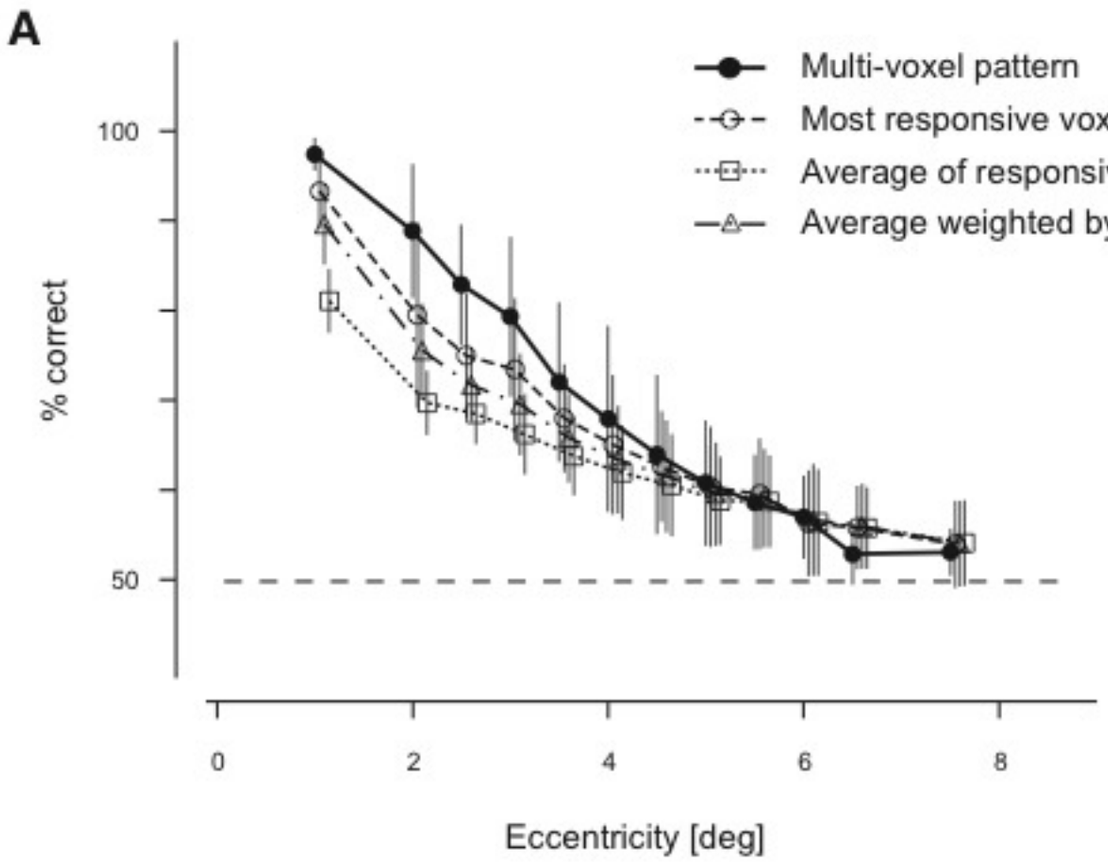


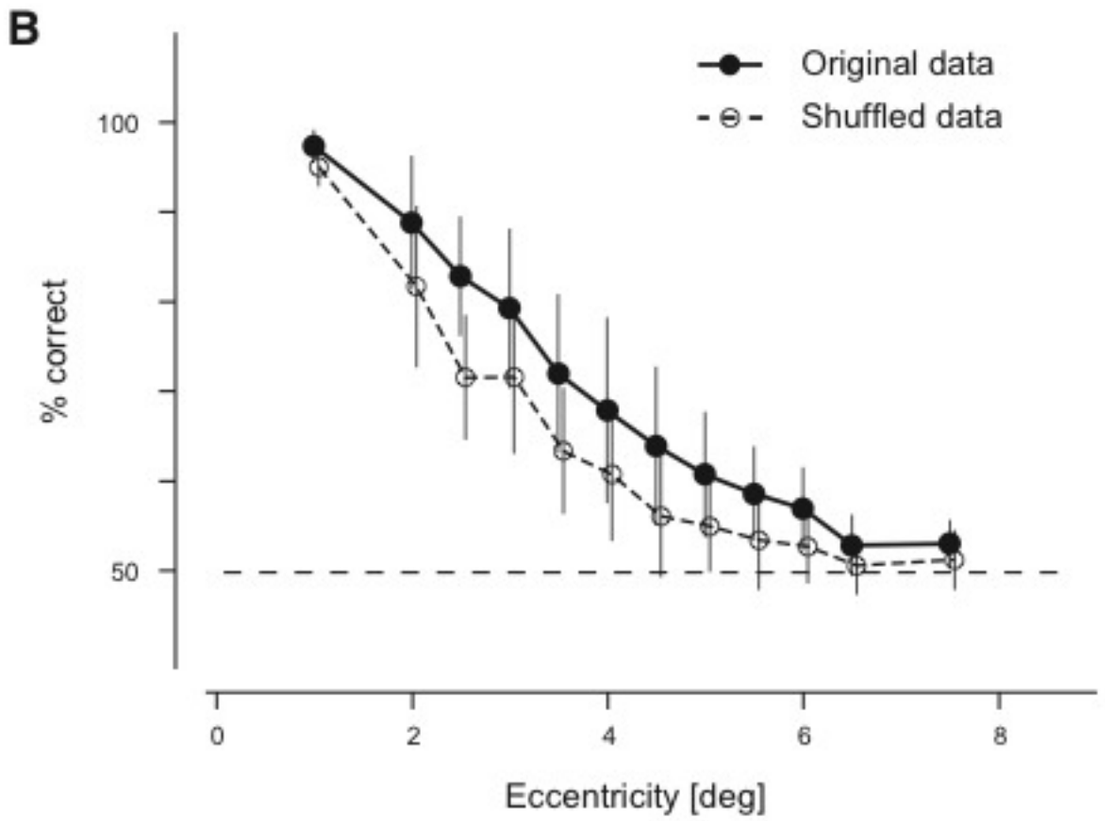




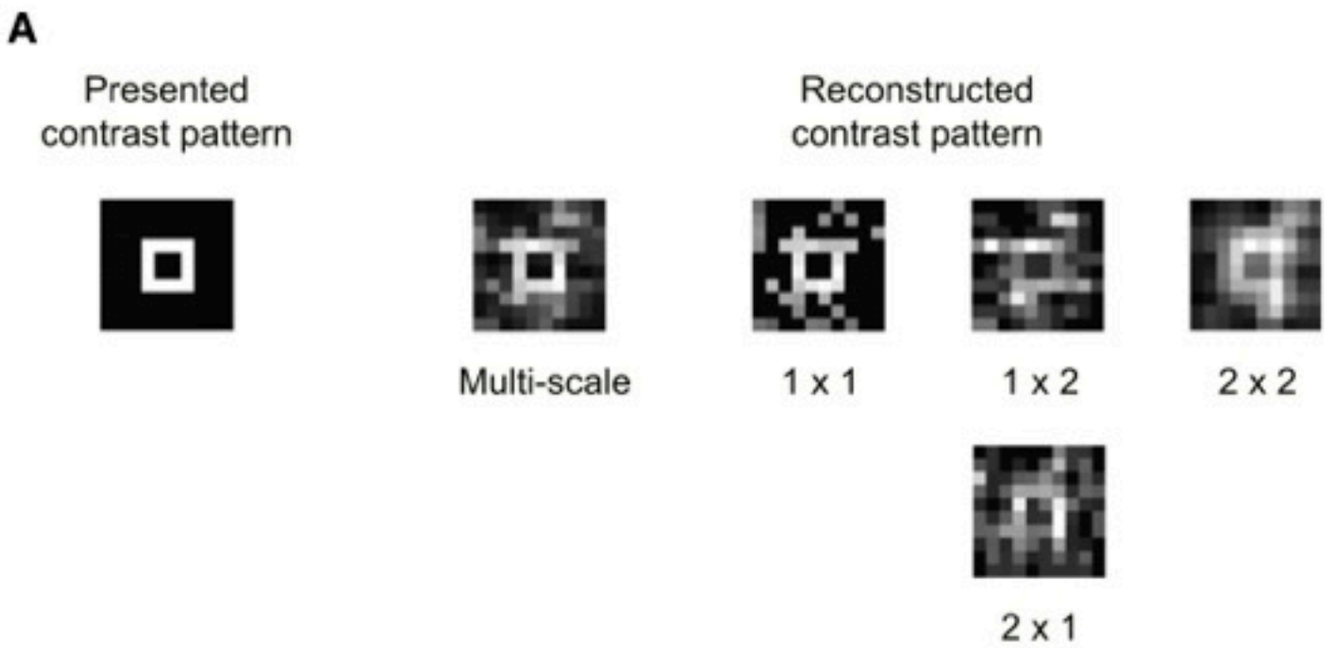


Multivoxel decoding

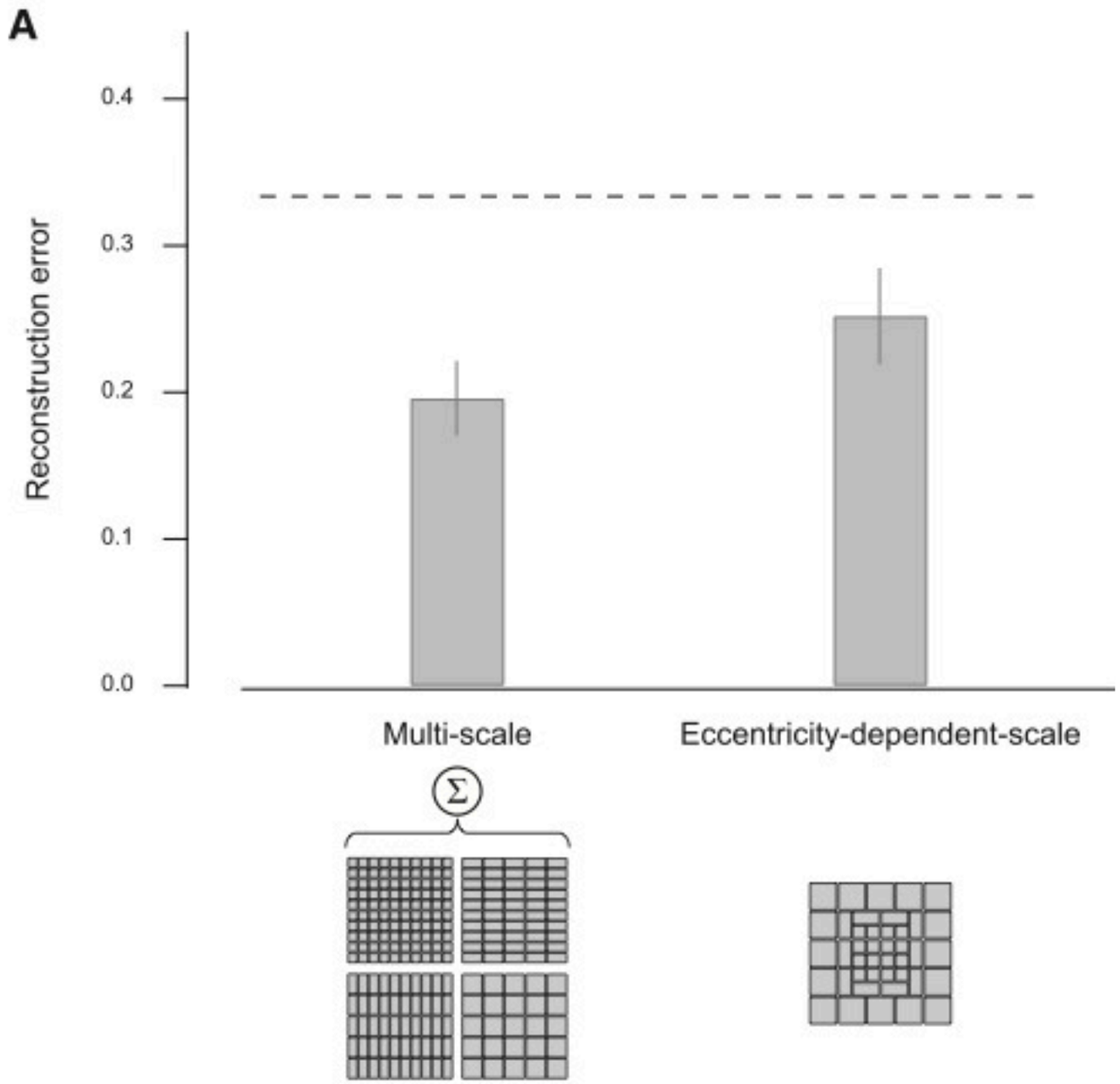


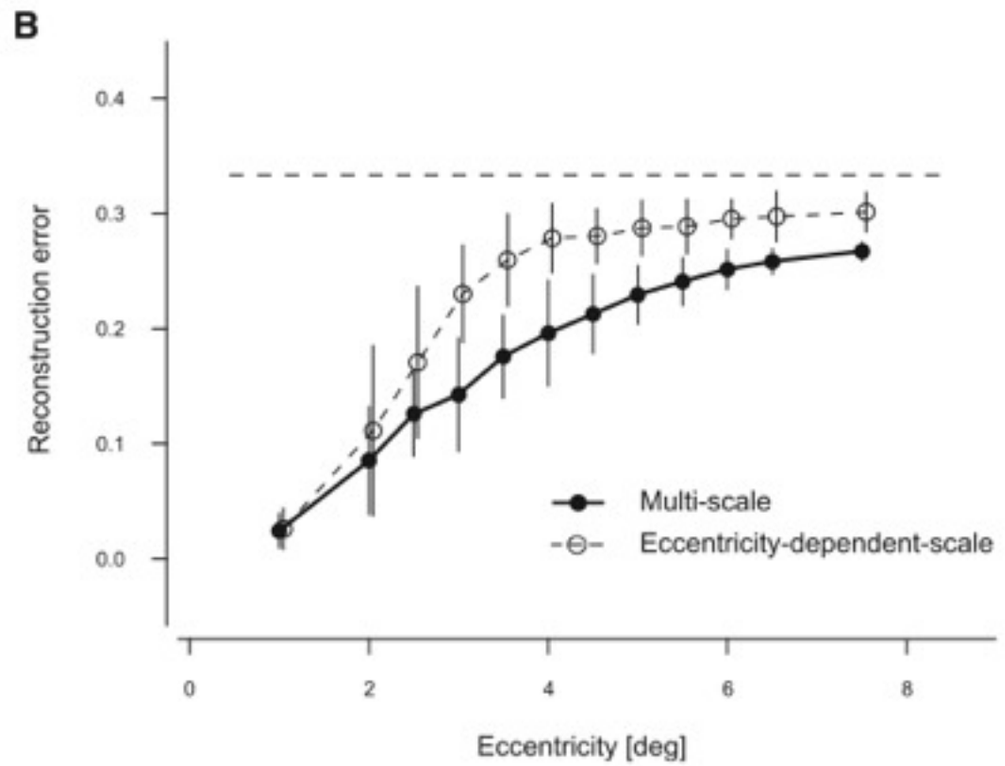


Multiscale decoding



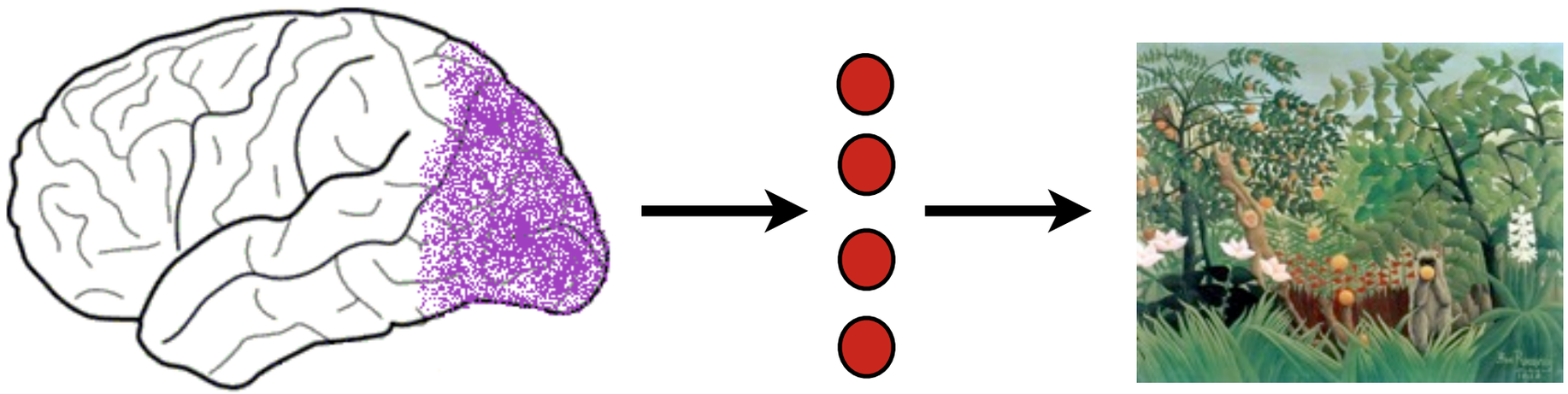
Multiscale decoding





Discriminative approach: sparse partial least squares

Predict image from a restricted set of responses
using a small number of latent variables

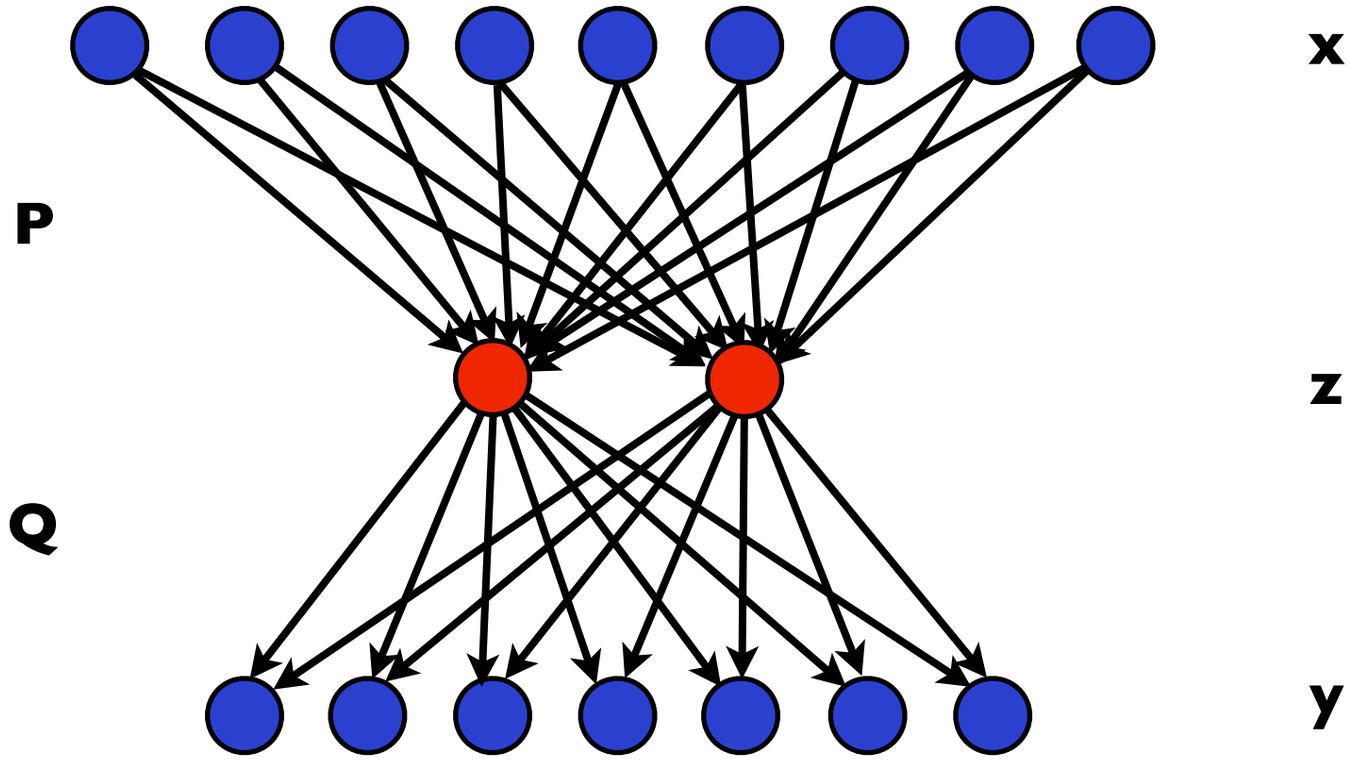


Key features:

- Linear: not enough data to (consistently) find strong nonlinear effects, stable, fast.
- Dimension reduction: gives a smooth image-like output, helps prevent overfitting.
- Sparsity: small number of relevant voxels makes the model interpretable.

van Gerven and Heskes. Sparse Orthonormalized Partial Least Squares. In: BNAIC. 2010.

Partial least squares

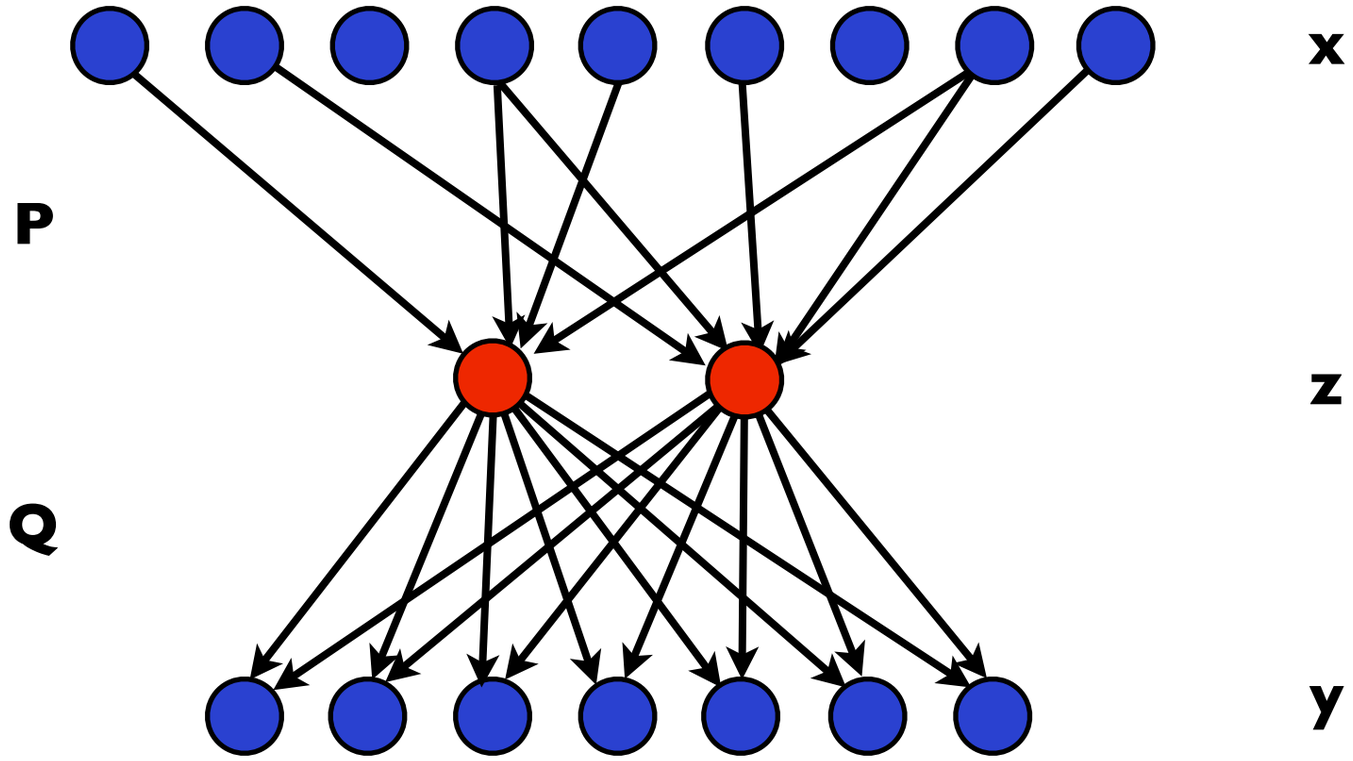


$$\begin{aligned} \mathbf{z} &= \mathbf{P}^\top \mathbf{x} \\ \mathbf{y} &= \mathbf{Q}\mathbf{z} \end{aligned}$$

Linear heteroencoder:

- Unique optimal solution (no local minima).
- reduces to principal component analysis in case x=y;
  rows of **Q** correspond to principal components of y.

Sparse partial least squares



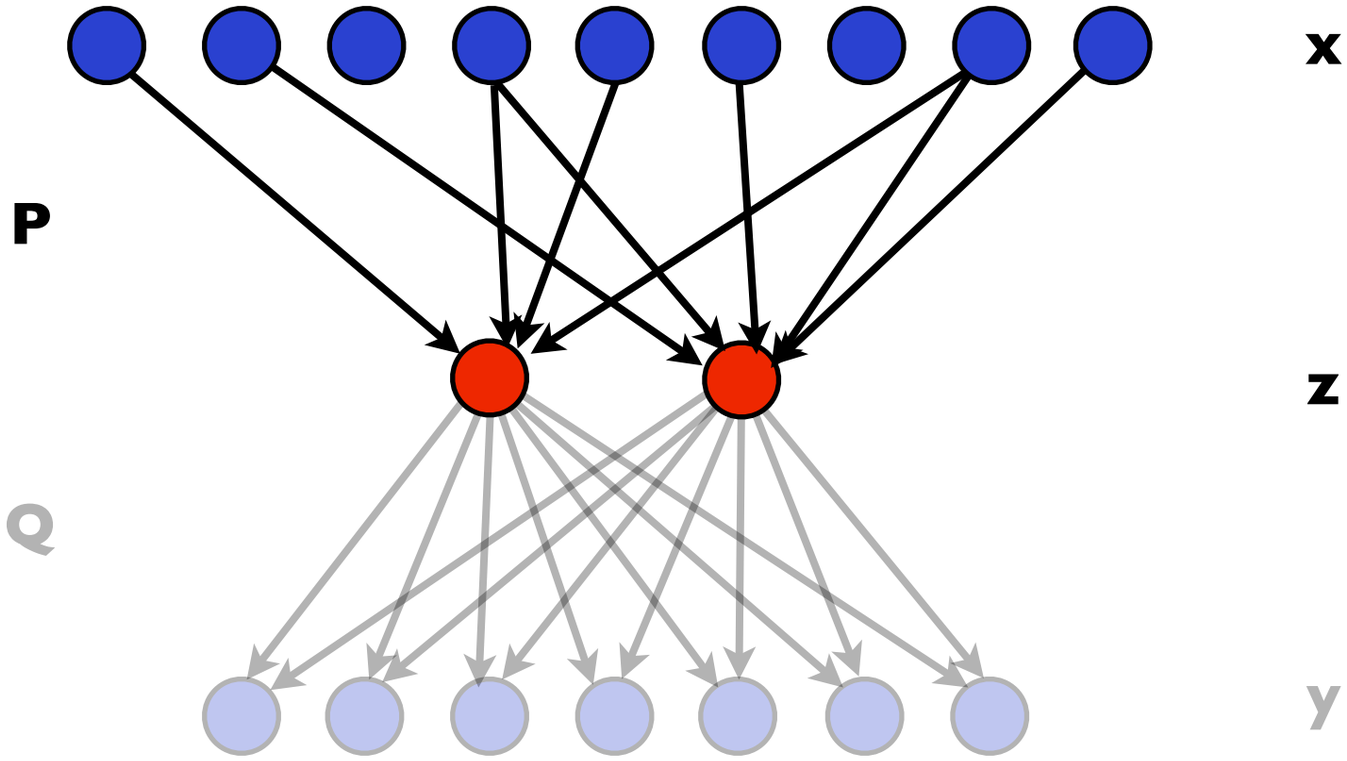
Objective:

$$(\hat{\mathbf{P}}, \hat{\mathbf{Q}}) = \arg\min_{\mathbf{P},\mathbf{Q}} \left[\frac{1}{2N} \sum_{n=1}^{N} ||\mathbf{y}^{(n)} - \mathbf{Q}\mathbf{P}^{\top}\mathbf{x}^{(n)}||_2^2 + R_{\nu,\Lambda}(\mathbf{P}) \right]$$

with $R_{\nu,\Lambda}(\mathbf{P}) = \nu \sum_{i=1}^{k} ||\mathbf{P}_i||_1 + \frac{1}{2} \sum_{j=1}^{k} \mathbf{P}_j^{\top} \Lambda \mathbf{P}_j$

- reduces to sparse PCA in case x=y (Zou et al., J Comput Graph Stat, 2006)

Iterative sparse PLS algorithm



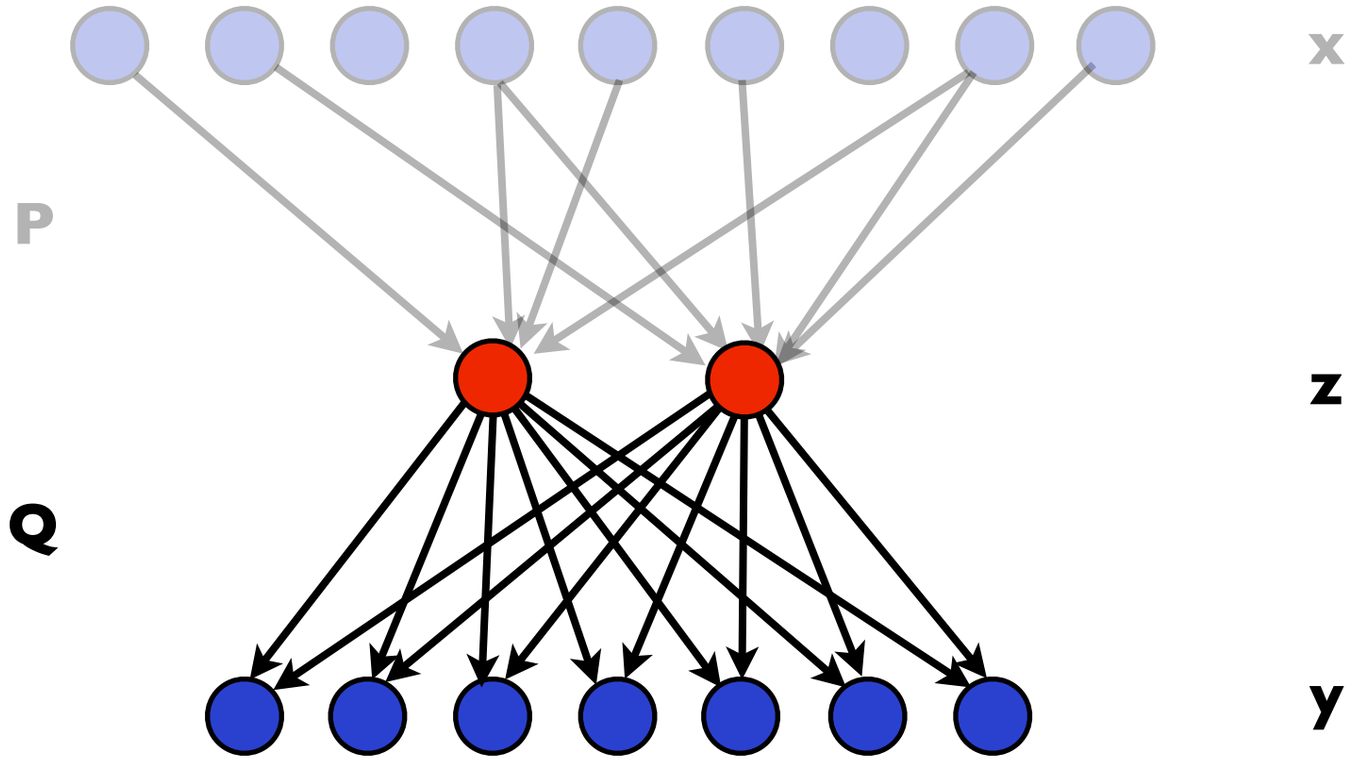
Fix $\mathbf{Q}$, reconstruct $\mathbf{Z} = \mathbf{Q}^T\mathbf{Y}$, and solve

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \left[\frac{1}{2N} \sum_{n=1}^{N} ||\mathbf{z}^{(n)} - \mathbf{P}^T\mathbf{x}^{(n)}||_2^2 + R_{\nu,\Lambda}(\mathbf{P}) \right]$$

- set of standard elastic net problems

Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J. Stat. Softw. 2010;33(1):1–22.

Iterative sparse PLS algorithm



Fix $\mathbf{P}$, reconstruct $\mathbf{Z} = \mathbf{P}^T\mathbf{X}$, and solve

$$\hat{\mathbf{Q}} = \arg\min_{\mathbf{P}} \left[\frac{1}{2N} \sum_{n=1}^{N} ||\mathbf{y}^{(n)} - \mathbf{Q}^T\mathbf{z}^{(n)}||_2^2 \right]$$

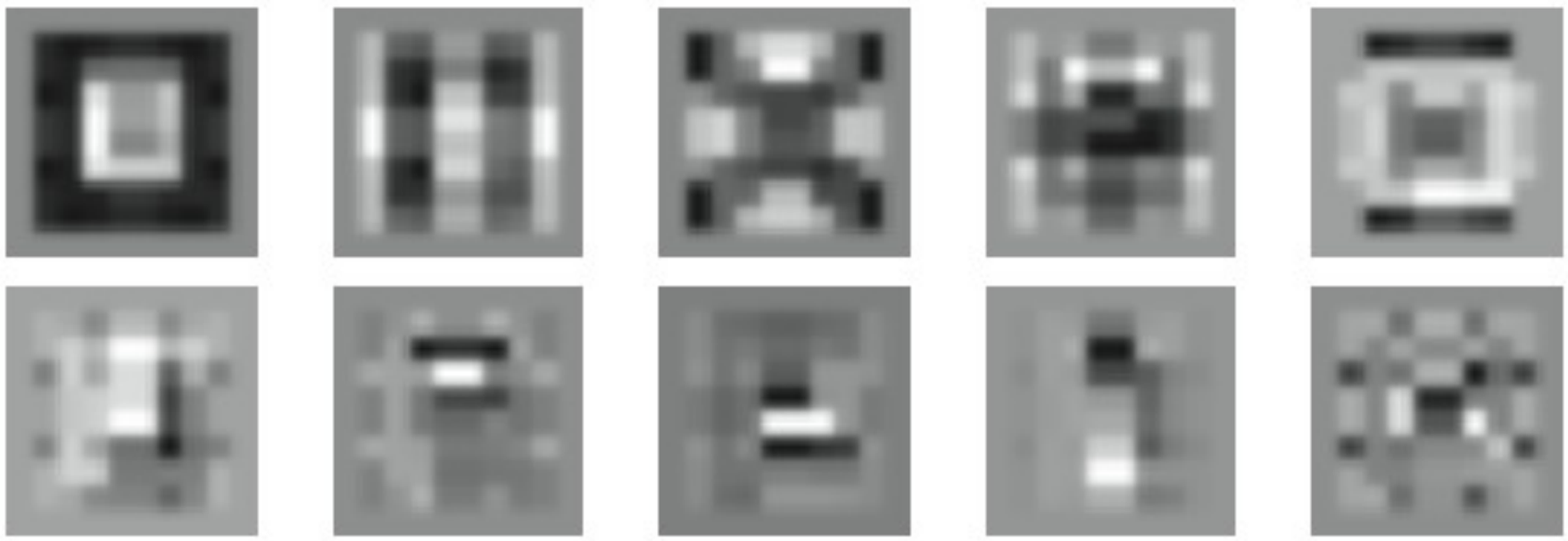
subject to $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_k$

- $\hat{\mathbf{Q}} = \boldsymbol{\Sigma}_{yz} \left(\boldsymbol{\Sigma}_{yz}^T \boldsymbol{\Sigma}_{yz} \right)^{-1/2}$ with $\boldsymbol{\Sigma}_{yz} \equiv \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}^{(n)} \left(\mathbf{z}^{(n)} \right)^T$

Experiment

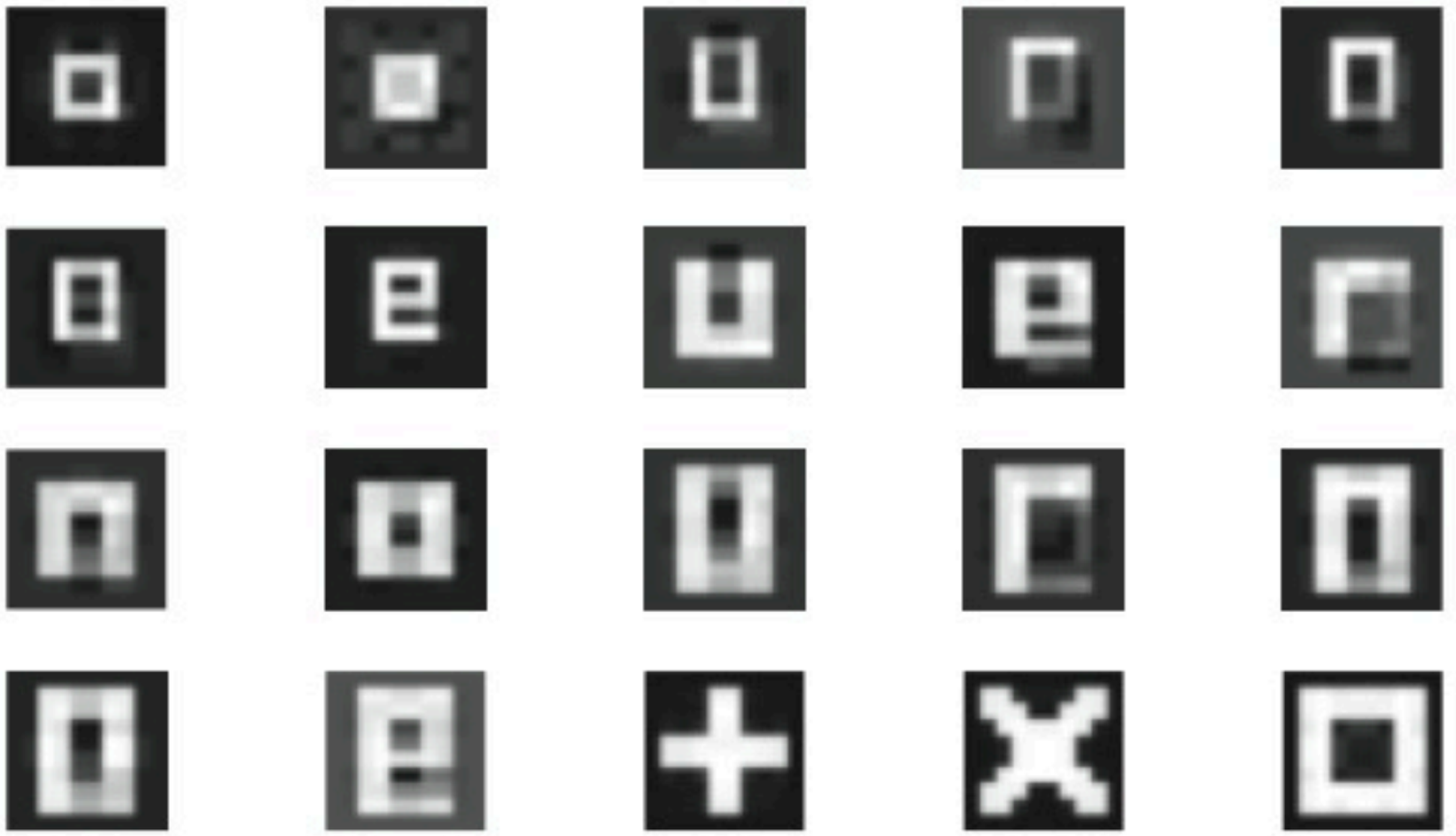
- Miyawaki et al., Neuron, 2008
- 10x10 images (geometric)
- BOLD response measured in 1017 voxels in primary visual cortex
- 10 latent variables, ν=0.01

Learned features



Learned features (rows of the matrix **Q**) are similar to principal components of the original images but change as a function of ν

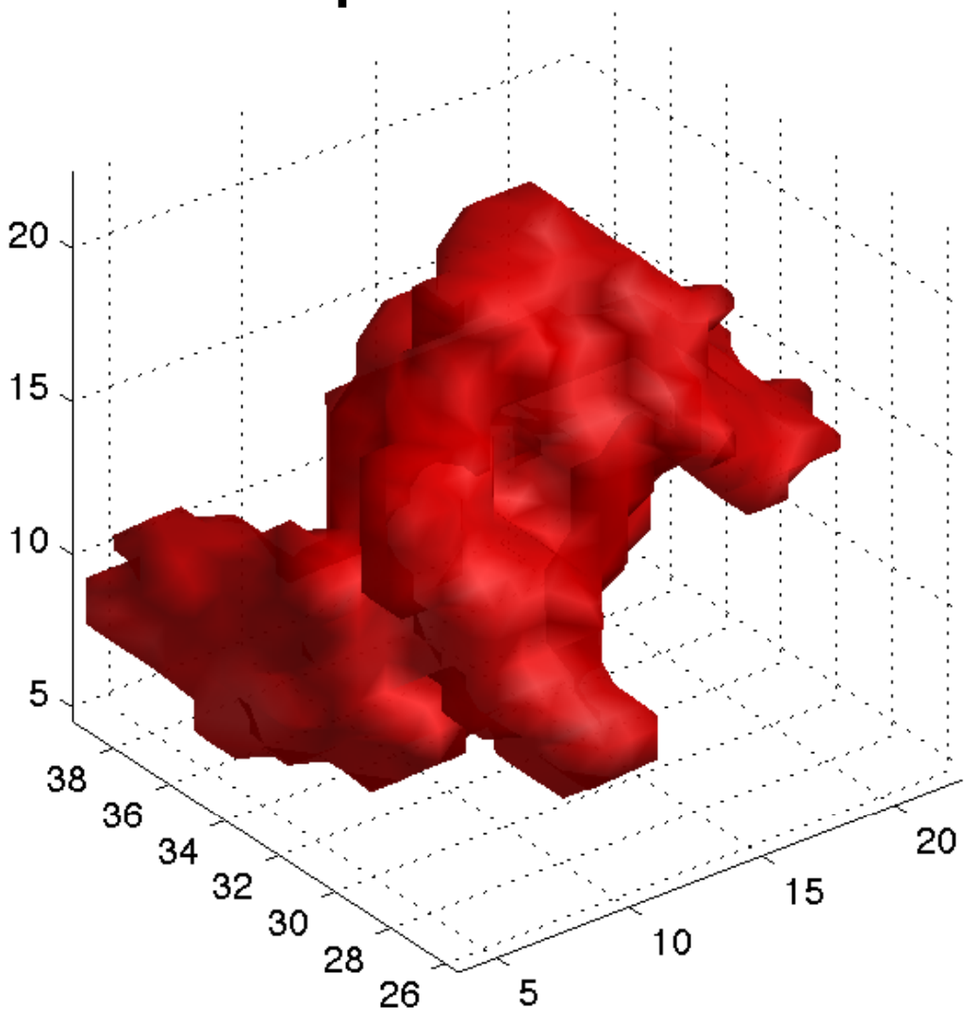
Reconstructions



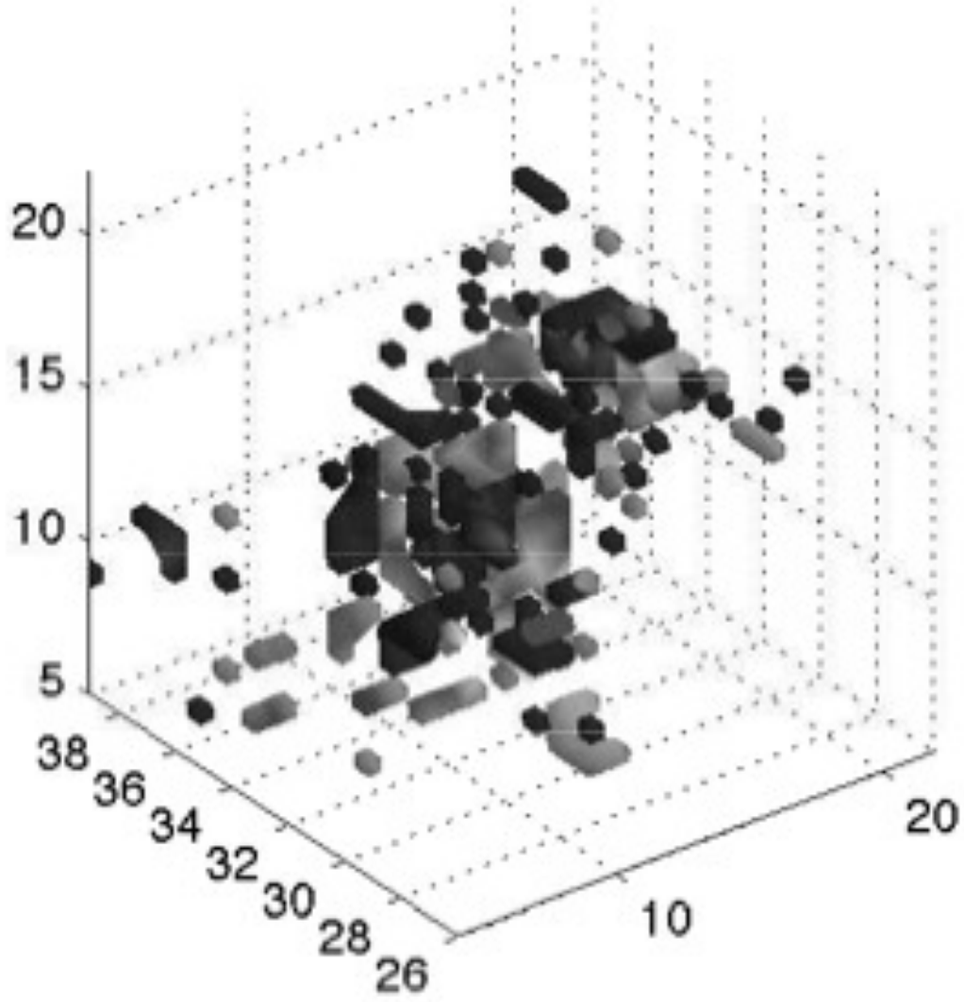
Reconstructions on hold-out data

selected voxels for the first latent variable out of all voxels in primary visual cortex

input voxels



selection



For $\nu = 0.01$, 80% of the parameters are set to zero

Other application: neuroprosthetics

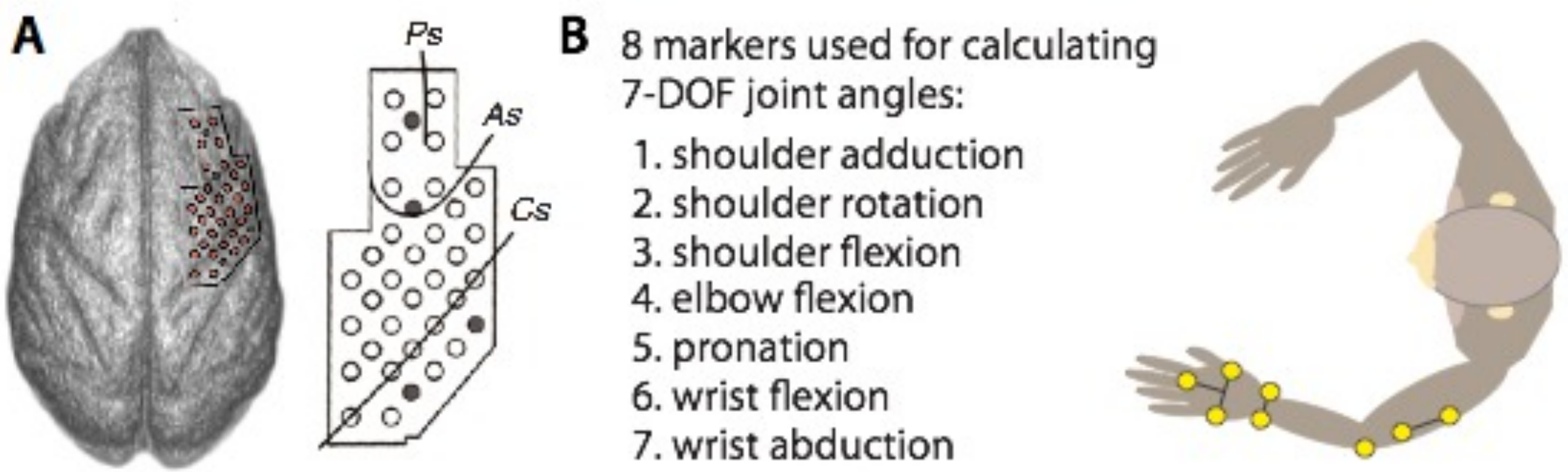
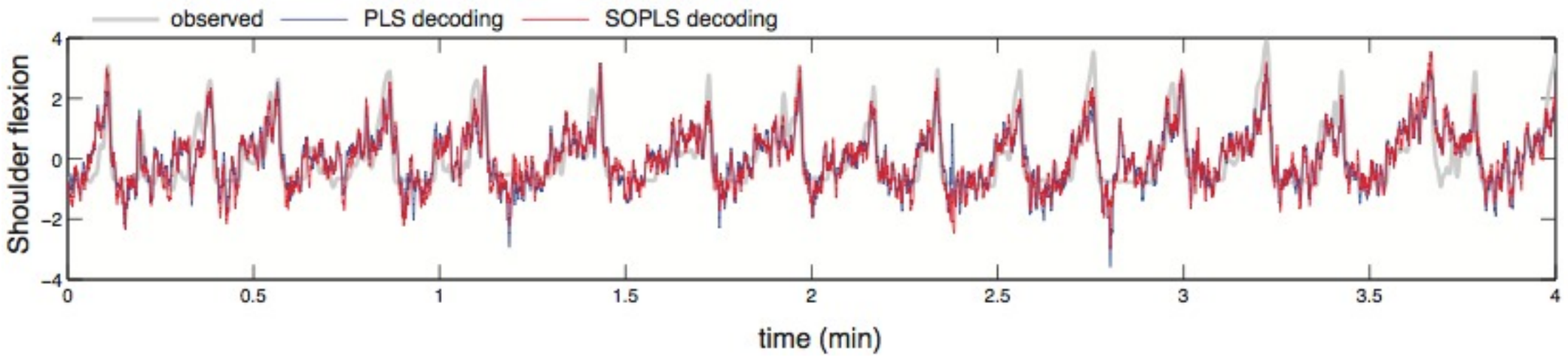
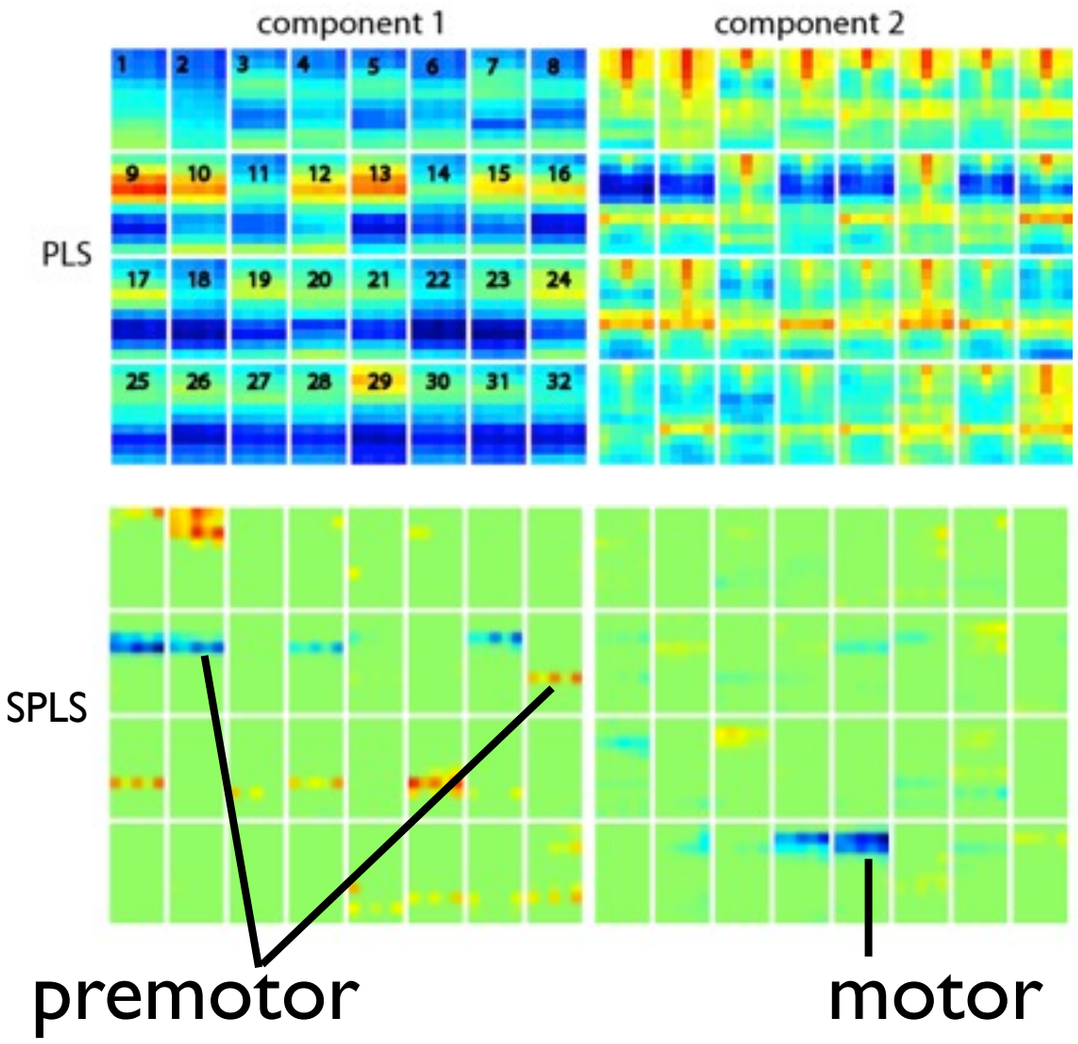


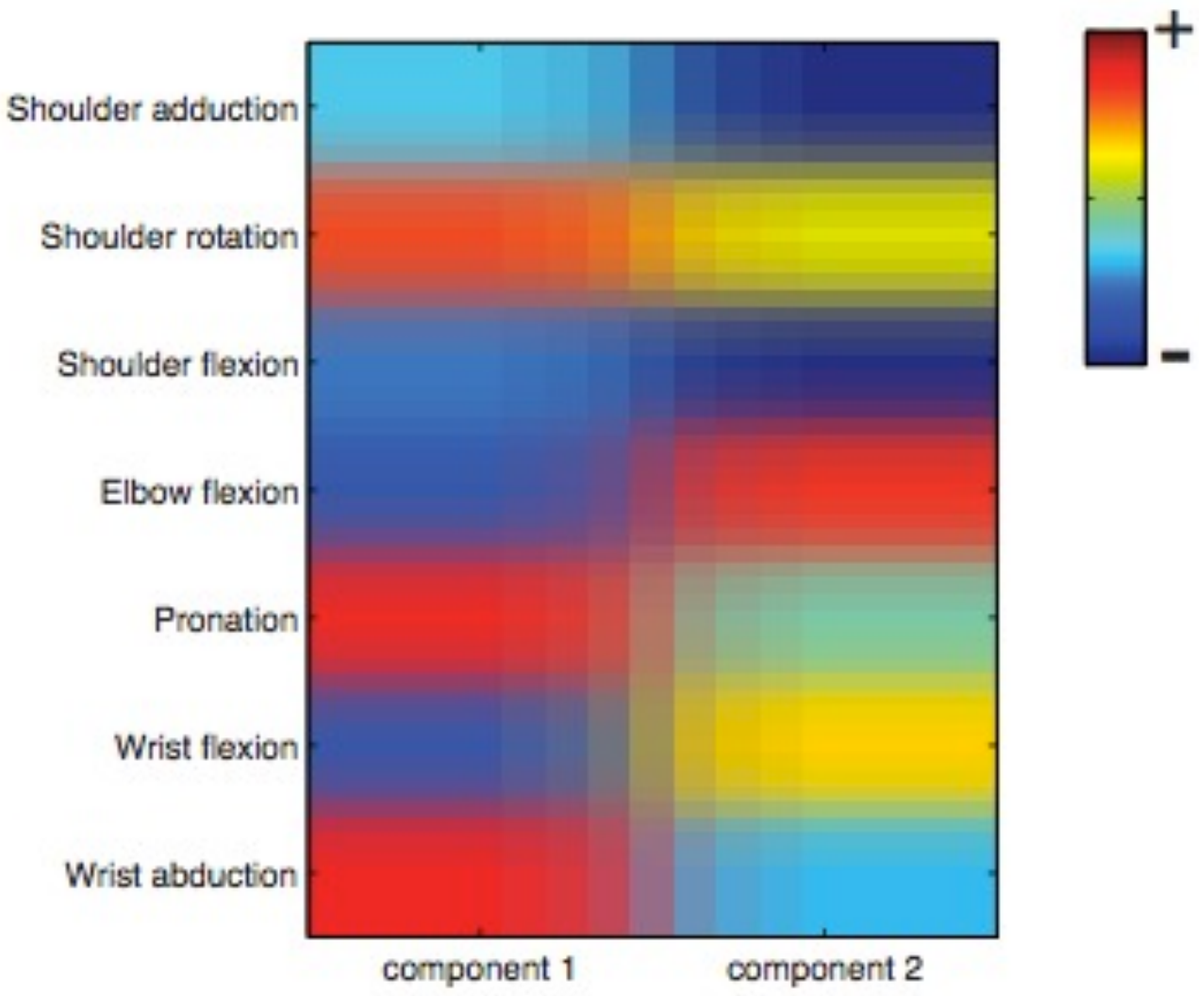
Figure 1: Experimental details. **A**. ECoG grid placement. **B**. Visualization of the markers used to compute the 7-DOF joint angles.

van Gerven MAJ, Chao ZC, Heskes T. On the Decoding of Intracranial Data using Sparse Orthonormalized Partial Least Squares. J Neural Eng. 2012 Feb. 23;9:026017.

Other application: neuroprosthetics







Required reading

- Miyawaki Y, Uchida H, Yamashita O, Sato M, Morito Y, Tanabe HC, et al. Visual Image Reconstruction from Human Brain Activity using a Combination of Multiscale Local Image Decoders. Neuron. 2008; 60(5):915–29.

Neuron
Article

Visual Image Reconstruction from Human Brain Activity using a Combination of Multiscale Local Image Decoders

Yoichi Miyawaki,[1,2,6] Hajime Uchida,[2,3,6] Okito Yamashita,[2] Masa-aki Sato,[2] Yusuke Morito,[4,5] Hiroki C. Tanabe,[4,5] Norihiro Sadato,[4,5] and Yukiyasu Kamitani[2,3,*]

Learning goals

- MAP vs MLE parameter estimation
- Feature selection, regularization (L1 and L2)
- Multiscale image decoding
- Idea of sparse partial least squares

Coming week

Practical

Continue tutorial

Lecture

Generative decoding