Voxelwise Modeling: understanding brain function with predictive models of brain activity

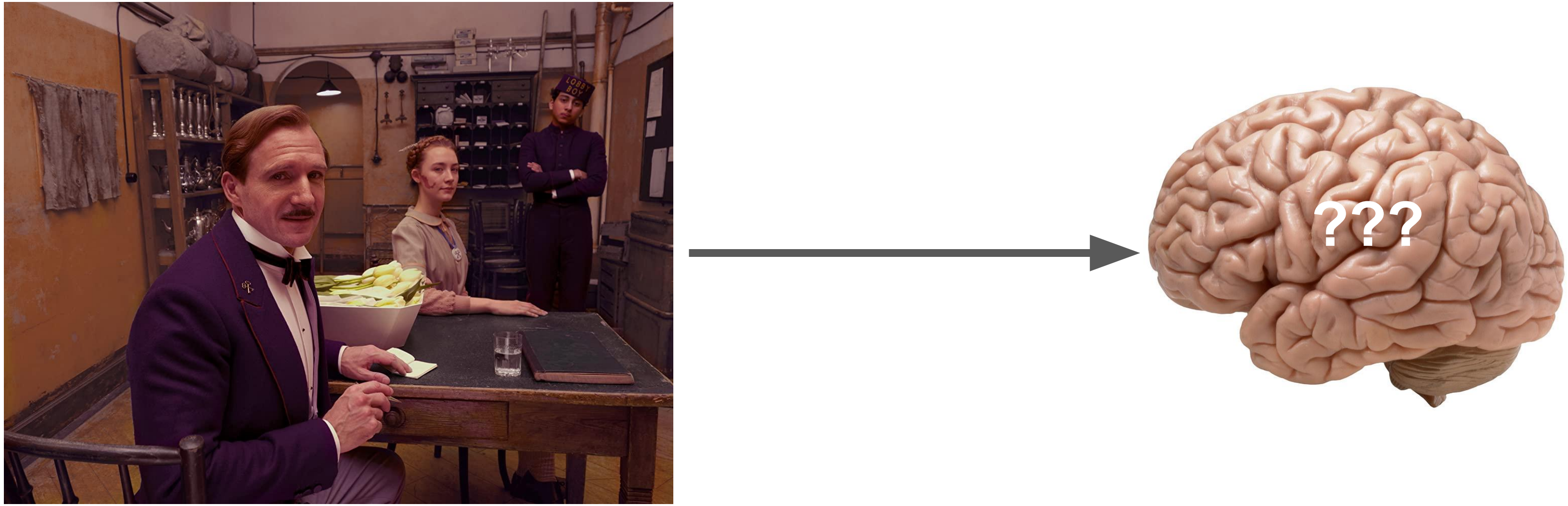
Matteo Visconti di Oleggio Castello

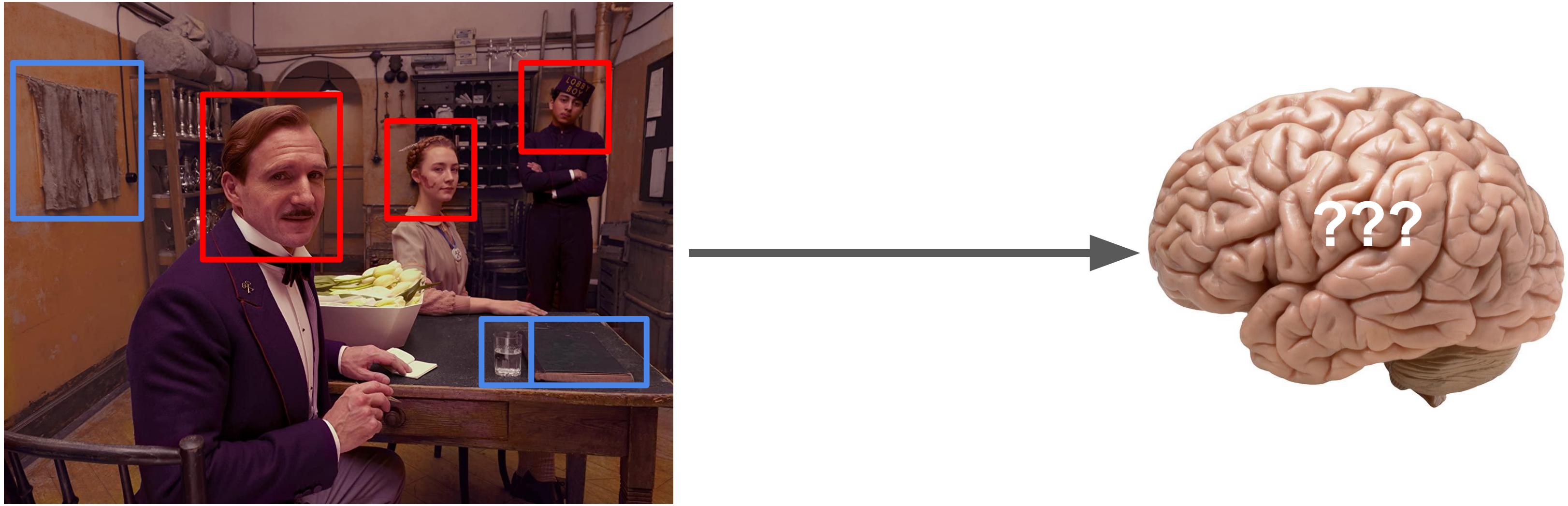
Tom Dupré la Tour

Gallant Lab

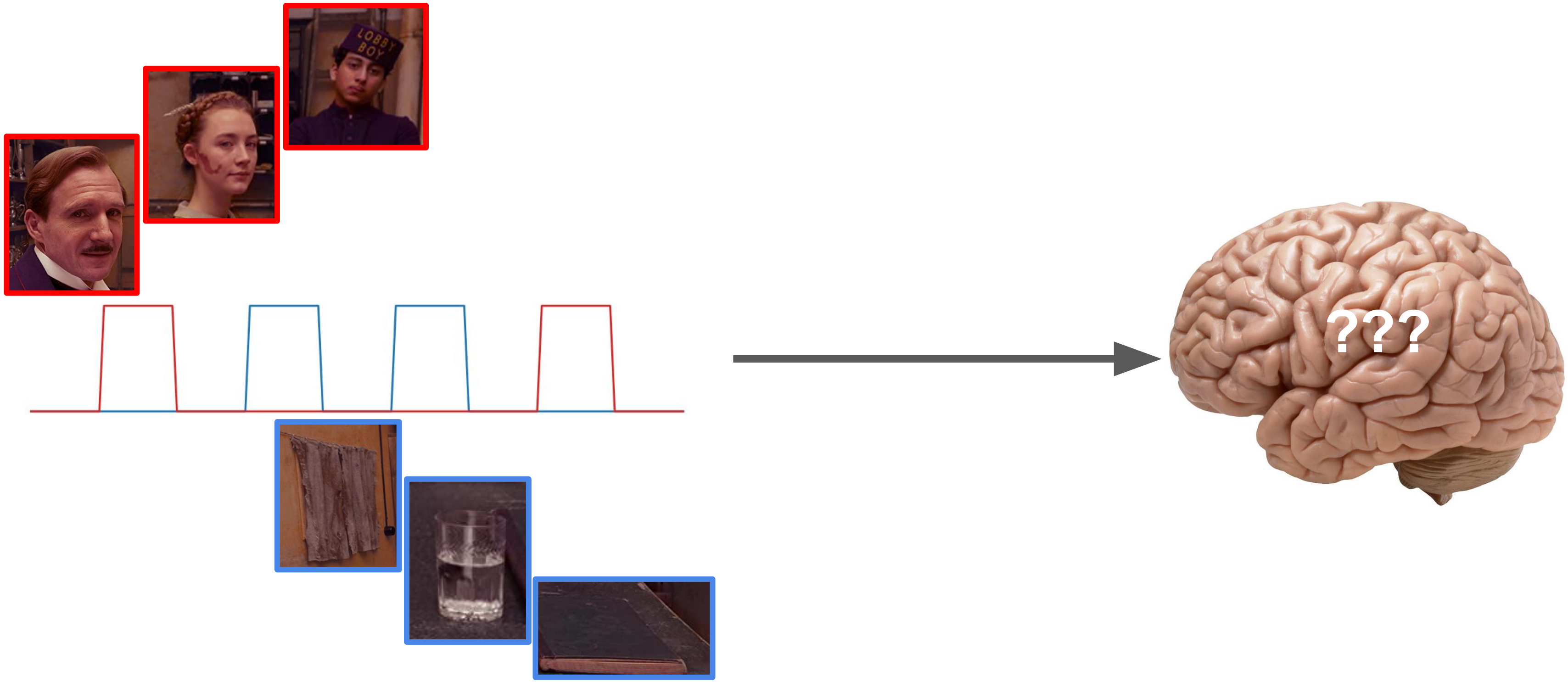
Cognitive Neuroscience Colloquium, UC Berkeley
March 8, 2021


LOBBY
BOY

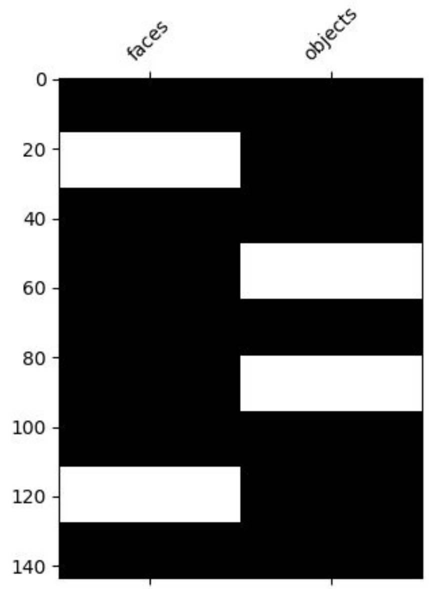

LOBBY BOY
???


LOBBY BOY
???

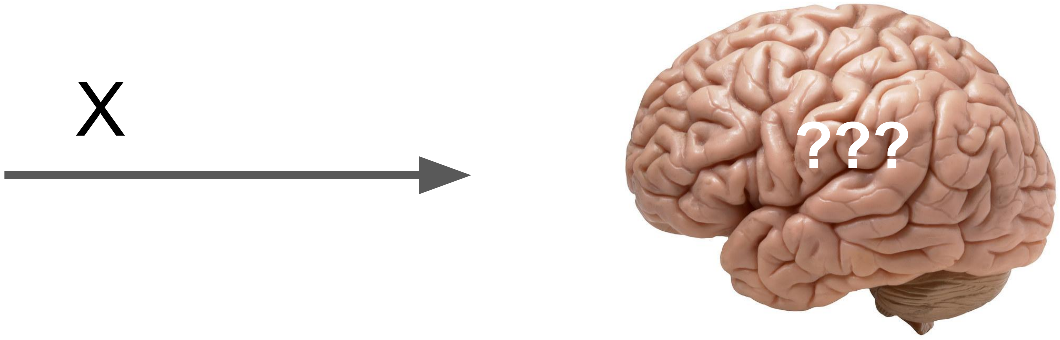
Classic GLM/SPM



Classic GLM/SPM



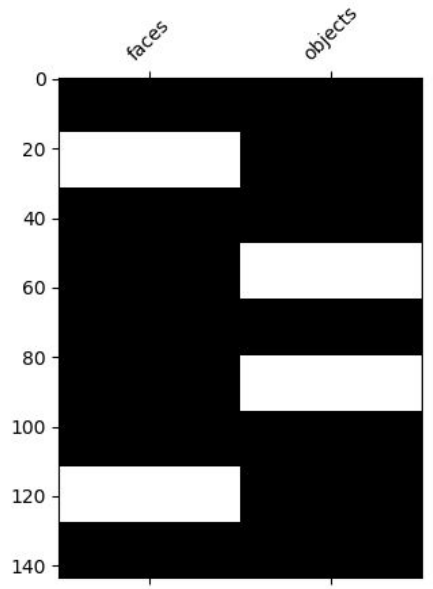
X



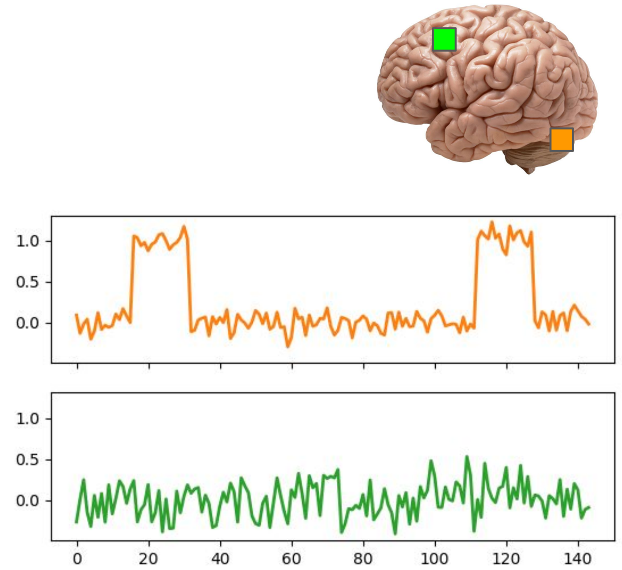
Not shown:

- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM



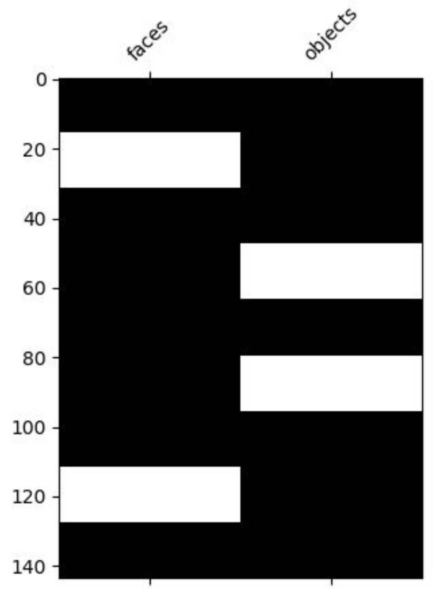
X Y



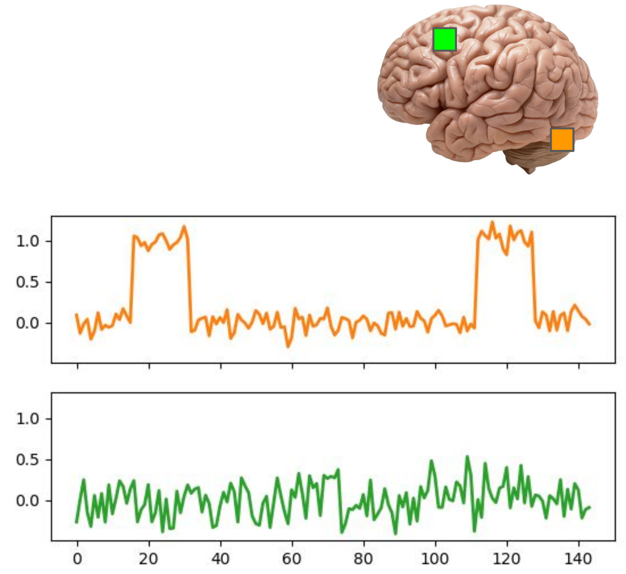
Not shown:

- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM



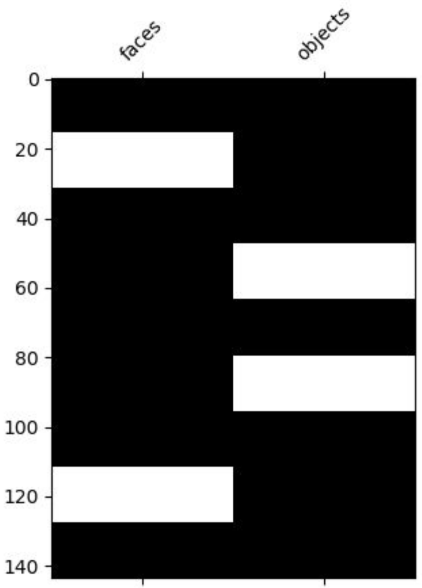
$$Xw = Y$$



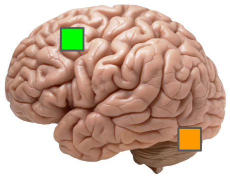
Not shown:

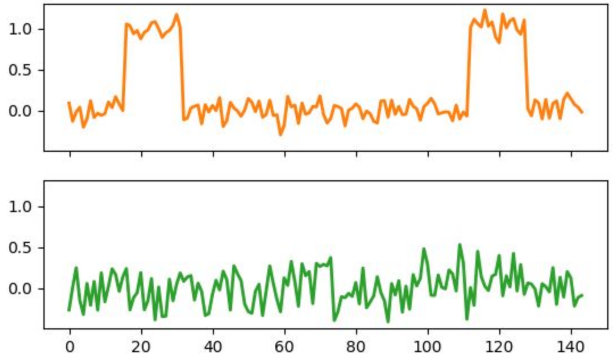
- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM



$$Xw = Y$$





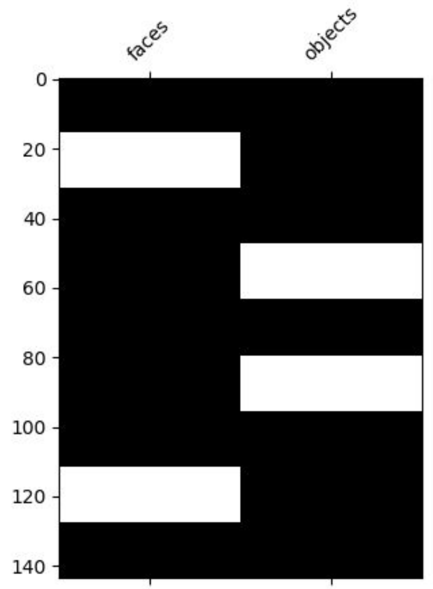
$w_1 = (0.9, 0)^T$

$w_2 = (0, 0)^T$

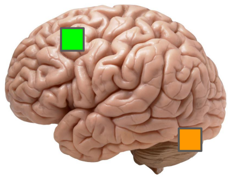
Not shown:

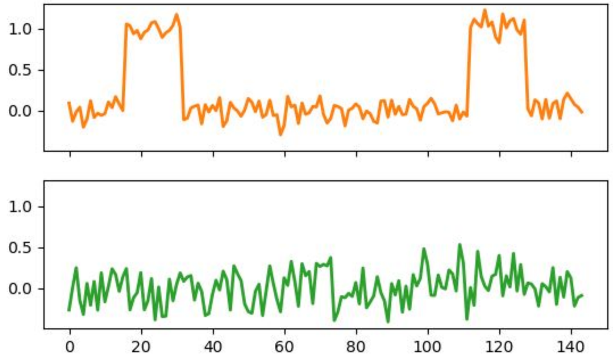
- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM



$$Xw = Y$$





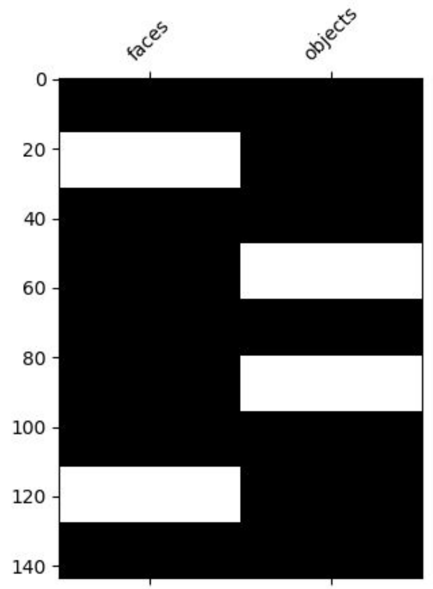
$w_1 = (0.9, 0)^T$ $SE_1 = (0.3, 0.9)^T$

$w_2 = (0, 0)^T$ $SE_2 = (0.6, 0.5)^T$

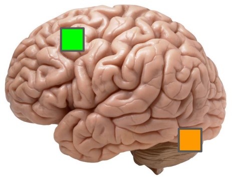
Not shown:

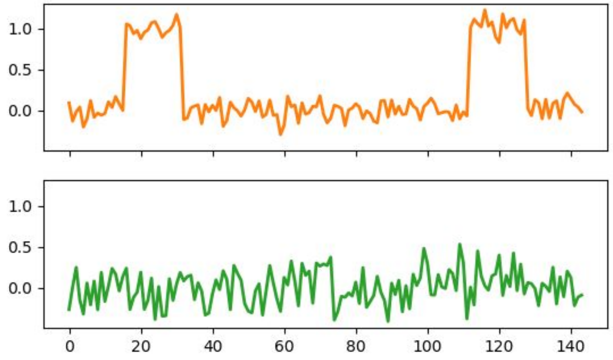
- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM



$$Xw = Y$$





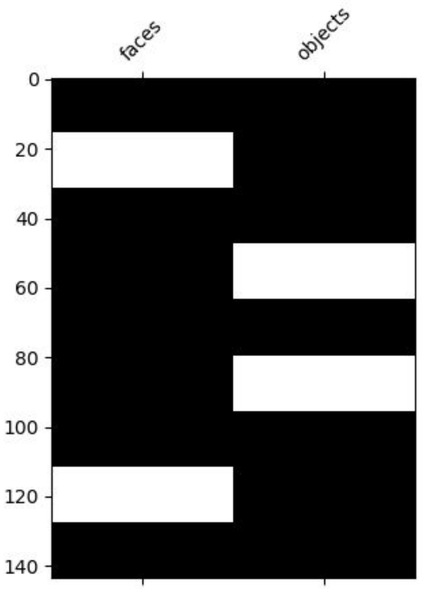
$w_1 = (0.9, 0)^T$ $SE_1 = (0.3, 0.9)^T$ $t_1 = (3, 0)^T$

$w_2 = (0, 0)^T$ $SE_2 = (0.6, 0.5)^T$ $t_2 = (0, 0)^T$

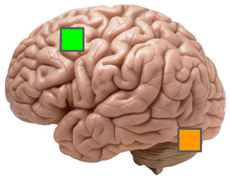
Not shown:

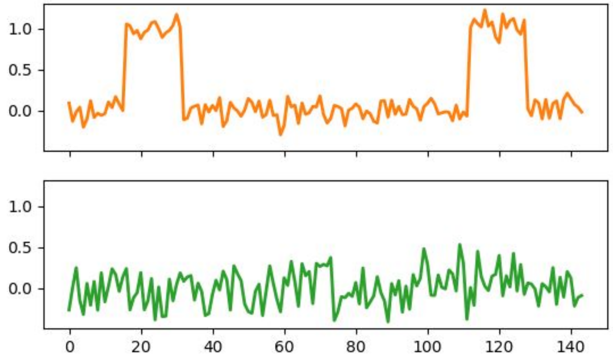
- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM



$$Xw = Y$$





$w_1 = (0.9, 0)^T$
$w_2 = (0, 0)^T$

Effect estimate

$SE_1 = (0.3, 0.9)^T$
$SE_2 = (0.6, 0.5)^T$

Noise estimate

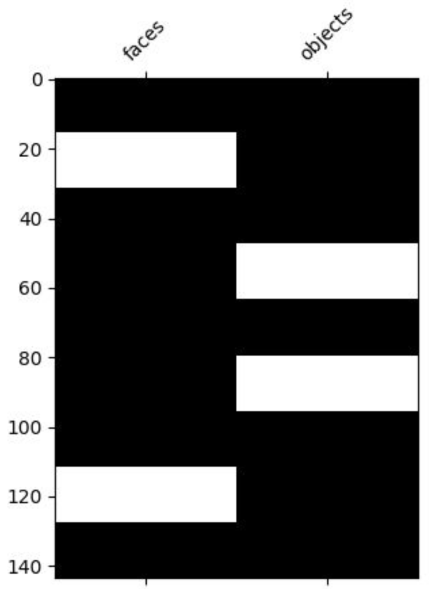
$t_1 = (3, 0)^T$
$t_2 = (0, 0)^T$

Statistic

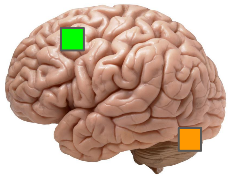
Not shown:

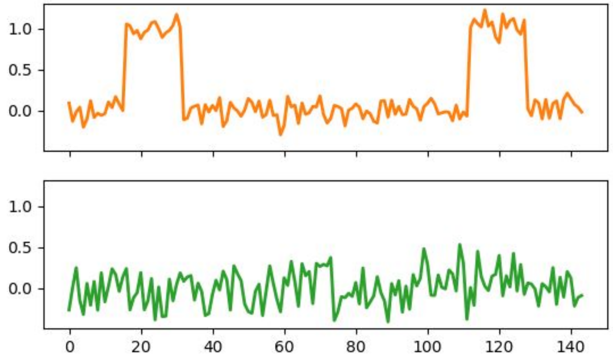
- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM



$$Xw = Y$$





$w_1 = (0.9, 0)^T$
$w_2 = (0, 0)^T$

Effect estimate

$SE_1 = (0.3, 0.9)^T$
$SE_2 = (0.6, 0.5)^T$

Noise estimate

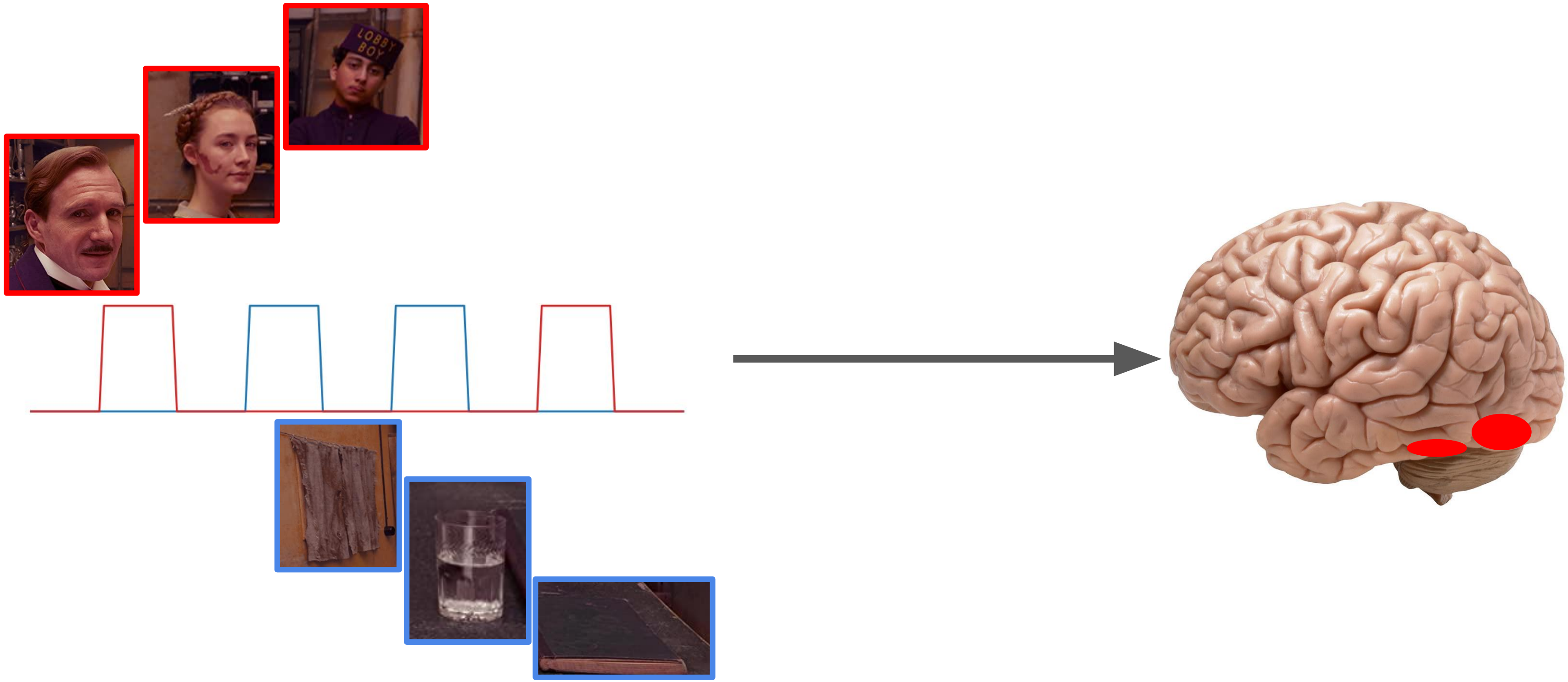
$t_1 = (3, 0)^T$
$t_2 = (0, 0)^T$

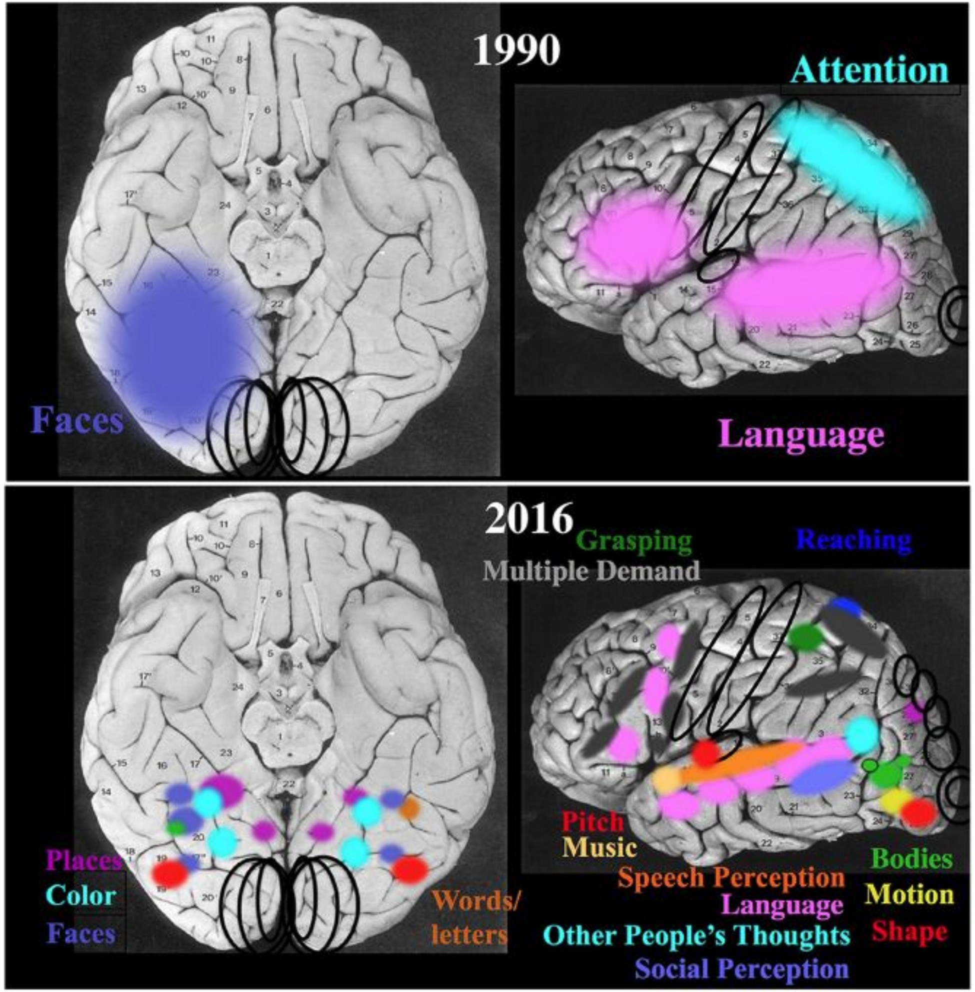
Statistic

Not shown:

- ways to account for HRF
- baseline
- nuisance regressors
- contrasts

Classic GLM/SPM

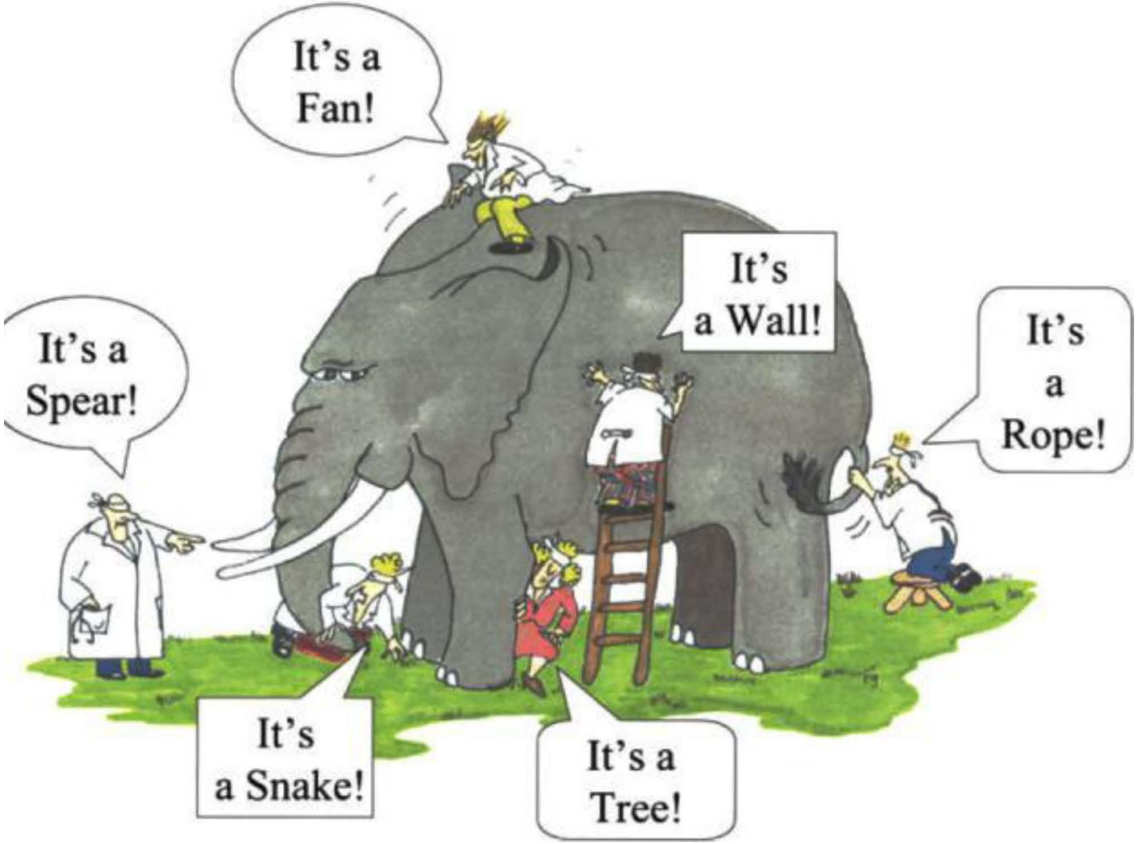




Kanwisher, 2017

Experimental problems with classic GLM/SPM

- Complex sensory and cognitive processes must be reduced to fit into designs that can be handled by an SPM approach
- Often this means simple factorial designs


It's a Fan!
It's a Spear!
It's a Wall!
It's a Rope!
It's a Snake!
It's a Tree!

Methodological problems with classic GLM/SPM

- Goodness-of-fit approach based on inferential statistics
    - Inferences are based on the significance of the estimated model parameters
    - Effect estimates are largely ignored (Chen, Taylor, & Cox, 2017)
        - statistical significance does not imply practical significance

Methodological problems with classic GLM/SPM

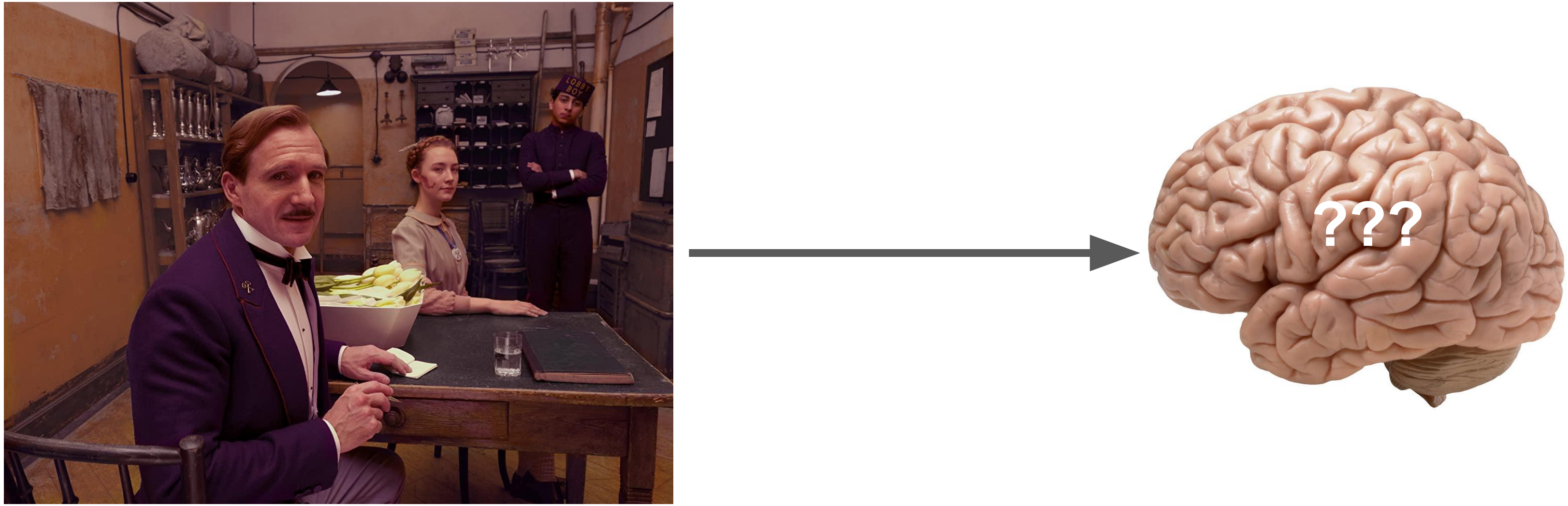
- Goodness-of-fit approach based on inferential statistics
  - Inferences are based on the significance of the estimated model parameters
  - Effect estimates are largely ignored (Chen, Taylor, & Cox, 2017)
    - statistical significance does not imply practical significance
- No measures of whether the results (and model parameters) will generalize to new conditions or datasets
  - models are fit in a single dataset (overfitting)
  - variance due to the (small number of) stimuli used is largely unaccounted for (stimulus-as-fixed-effect fallacy; Westfall, Nichols, & Yarkoni, 2017)

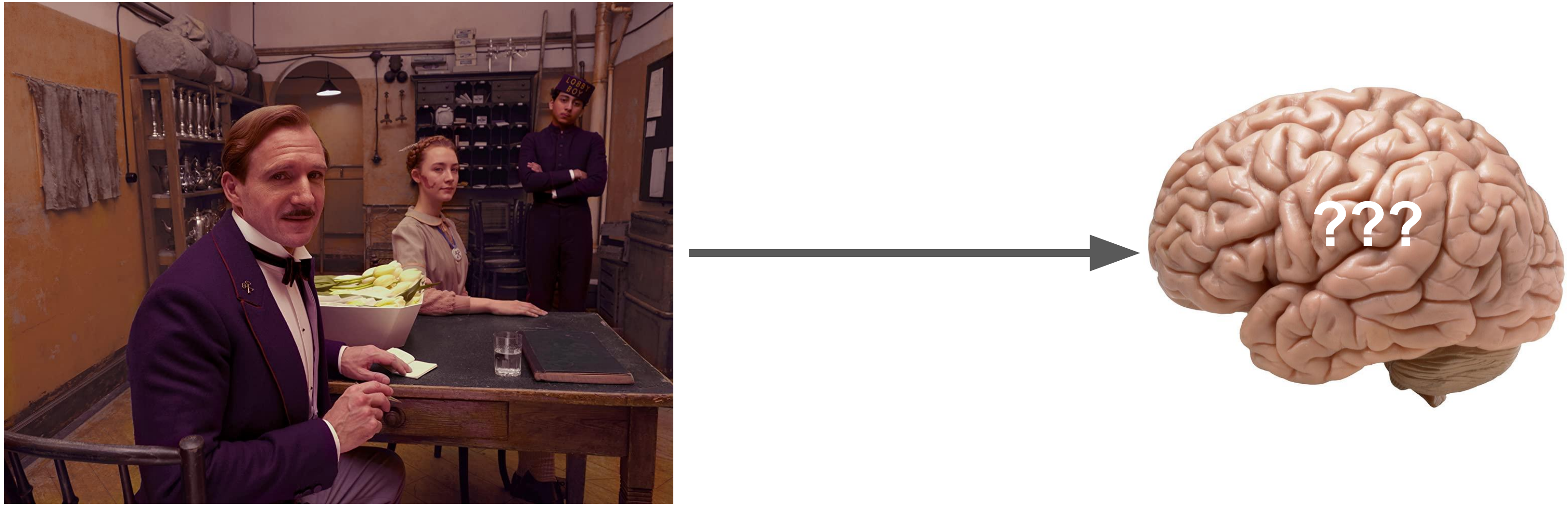
Methodological problems with classic GLM/SPM

- Classic GLM/SPM provides little guarantee that
  - the experimental results will replicate (Szucs & Ioannidis, 2017)
  - the model tested will generalize (Yarkoni, 2019; Westfall, Nichols, & Yarkoni, 2017)

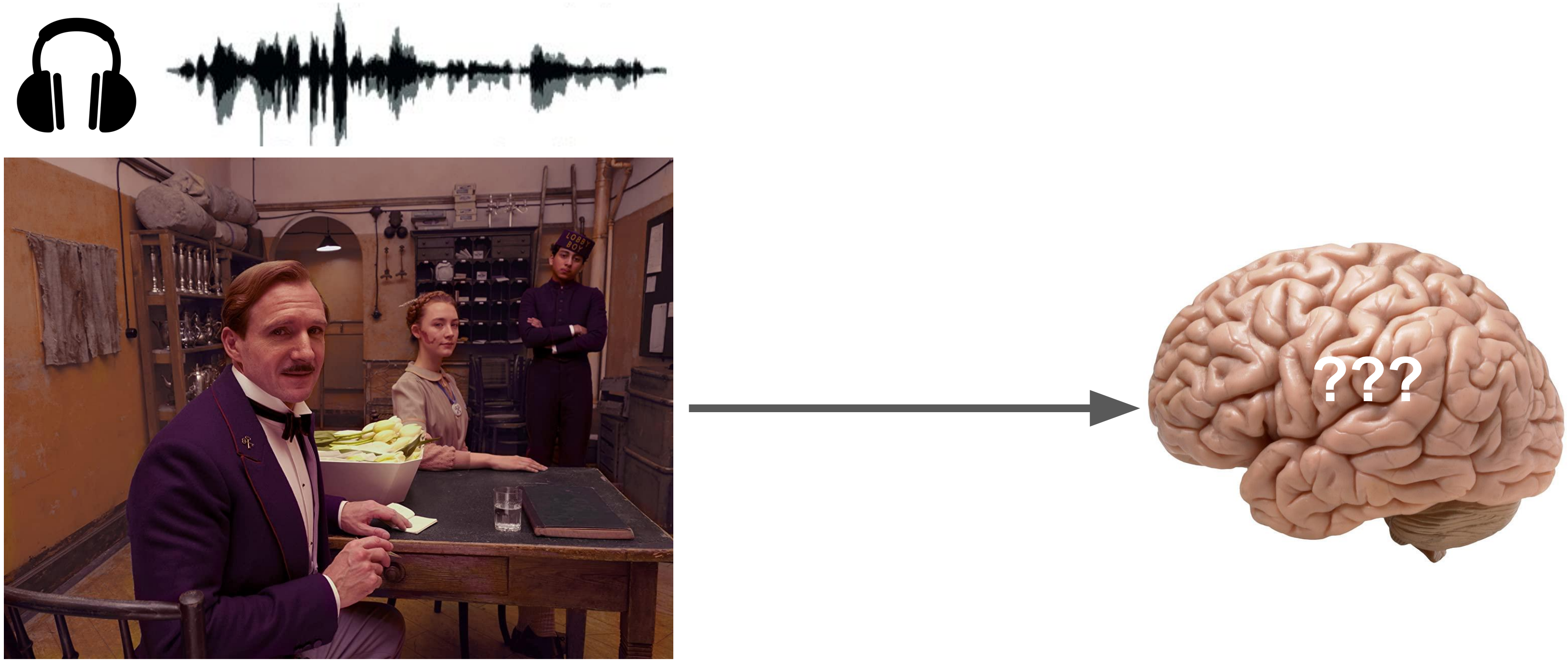
A different approach: Voxelwise Modeling

- Respect the complexity of the real world (do not reduce the elephant!)
- Avoid the goodness-of-fit approach and null-hypothesis statistical testing (*data modeling culture*; Breiman, 2001)
- Use methods from machine learning and data science (*algorithmic modeling culture*; Breiman, 2001)
  - Create models that accurately predict brain activity
  - Estimate **model prediction accuracy** on an **independent dataset**



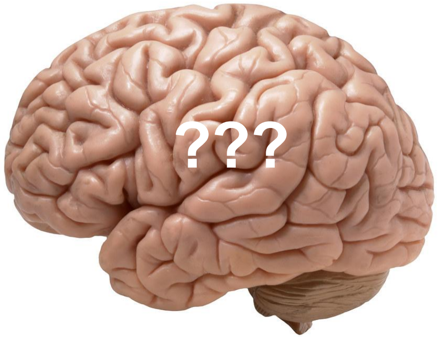


- low-level visual features (motion energy)
- objects in the scene
- facial expressions
- emotions portrayed
- social interactions



- low-level visual features (motion energy)
- objects in the scene
- facial expressions
- emotions portrayed
- social interactions

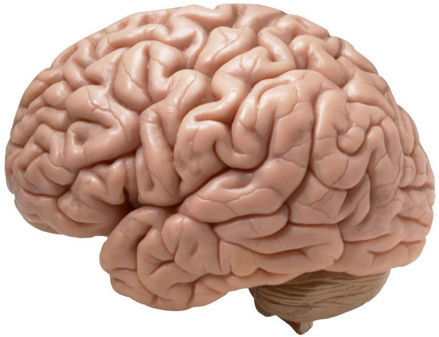
- spectral features
- speech content



- low-level visual features (motion energy)
- objects in the scene
- facial expressions
- emotions portrayed
- social interactions

- spectral features
- speech content

$$Xw = Y$$



- low-level visual features (motion energy)
- objects in the scene
- facial expressions
- emotions portrayed
- social interactions

$Xw = Y$

$$Xw = Y$$

Z

$$Xw = Y$$

$$Zw$$

$Xw = Y$

???

Zw

Model selection (training set)

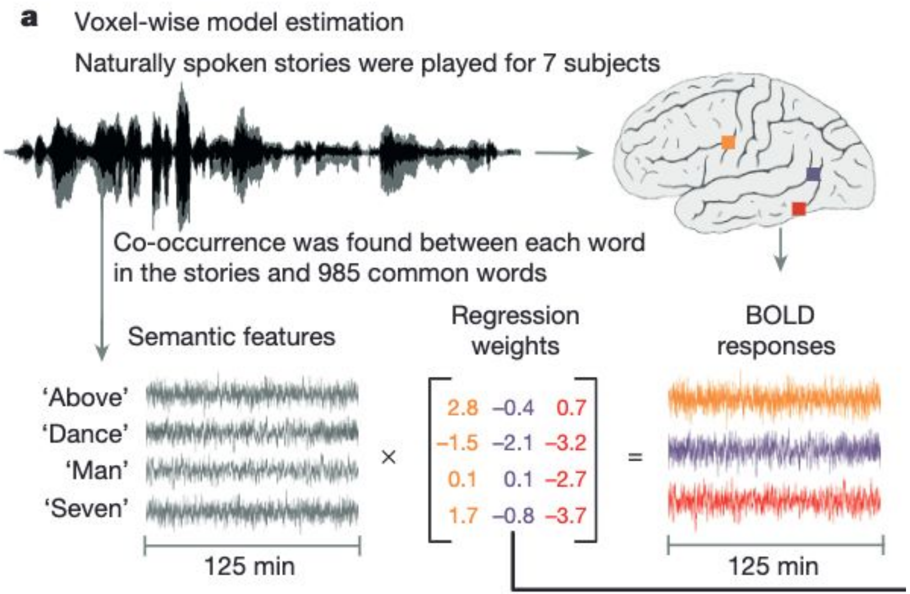
$Xw = Y$

???

Model assessment (test set)

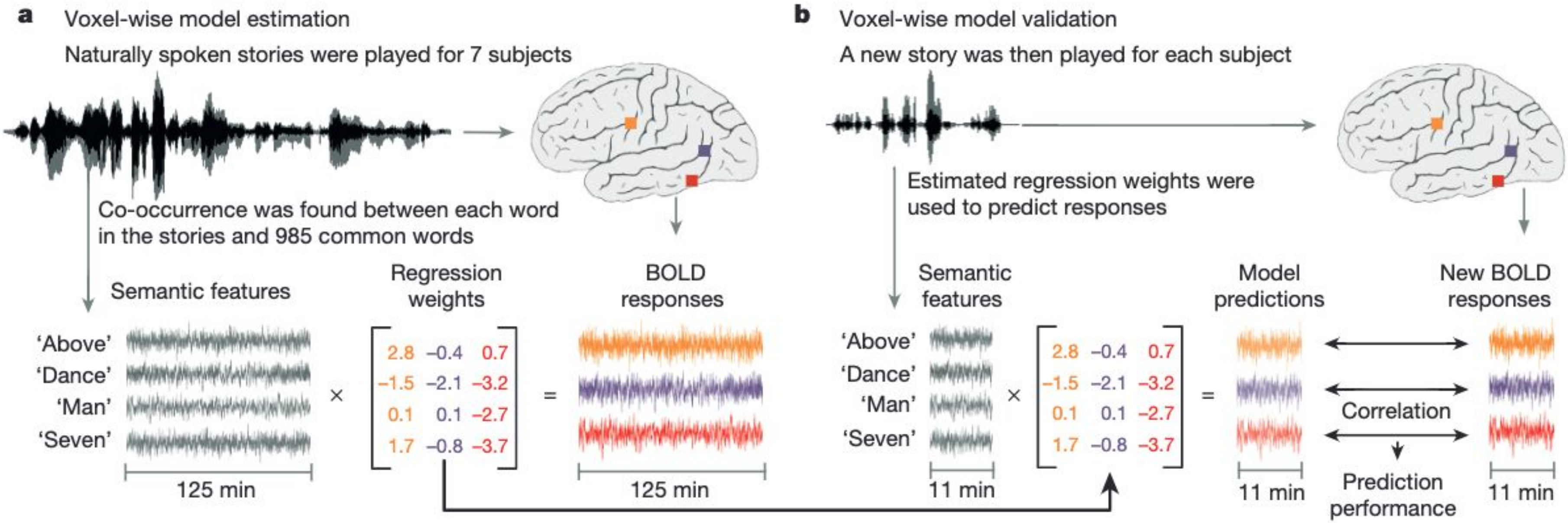
Zw

Example: Huth et al., 2016



Model selection
(Training set)

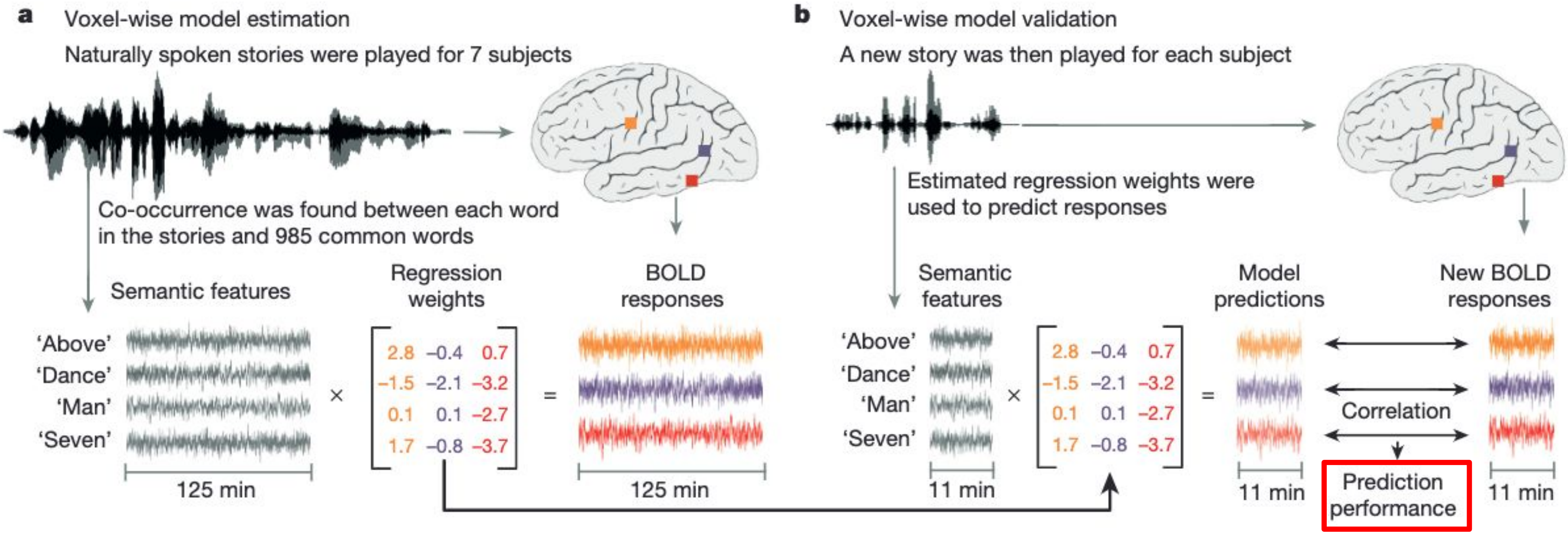
Example: Huth et al., 2016



Model selection
(Training set)

Model assessment
(Test set)

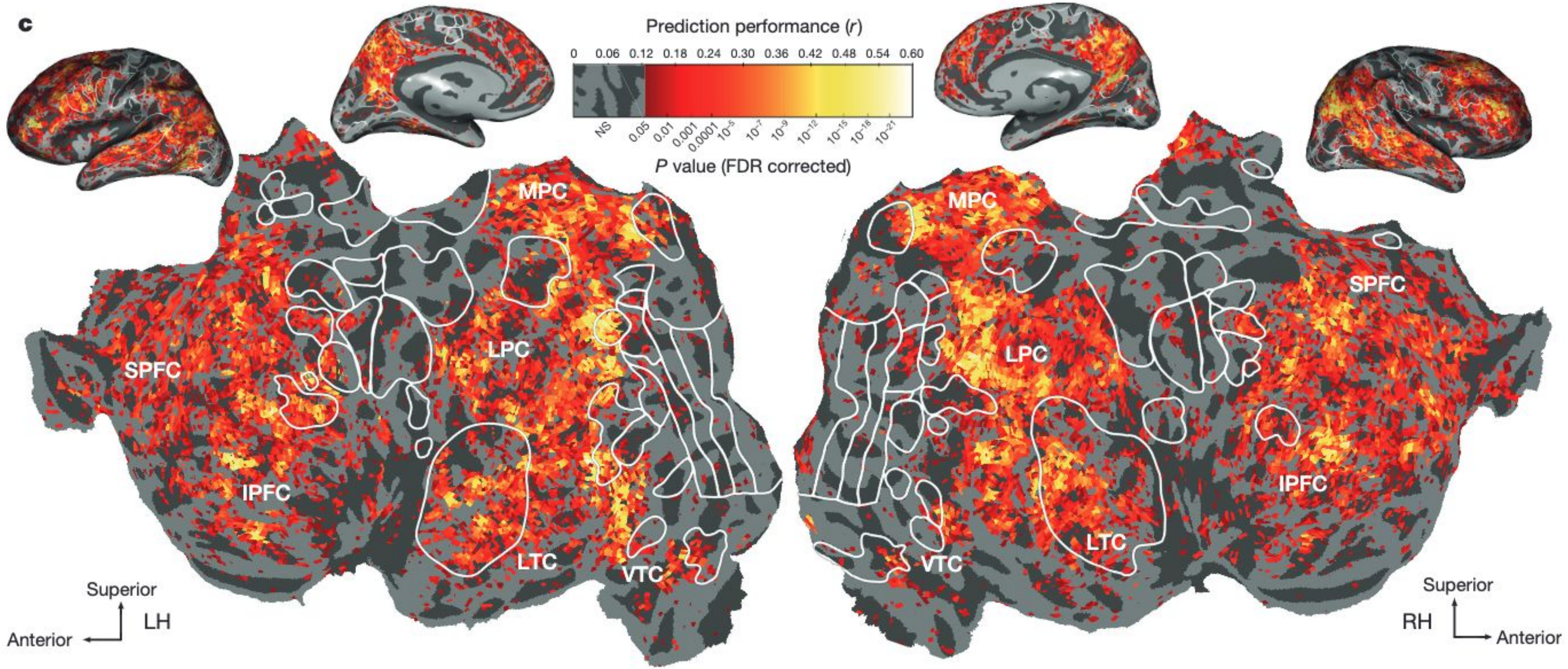
Example: Huth et al., 2016



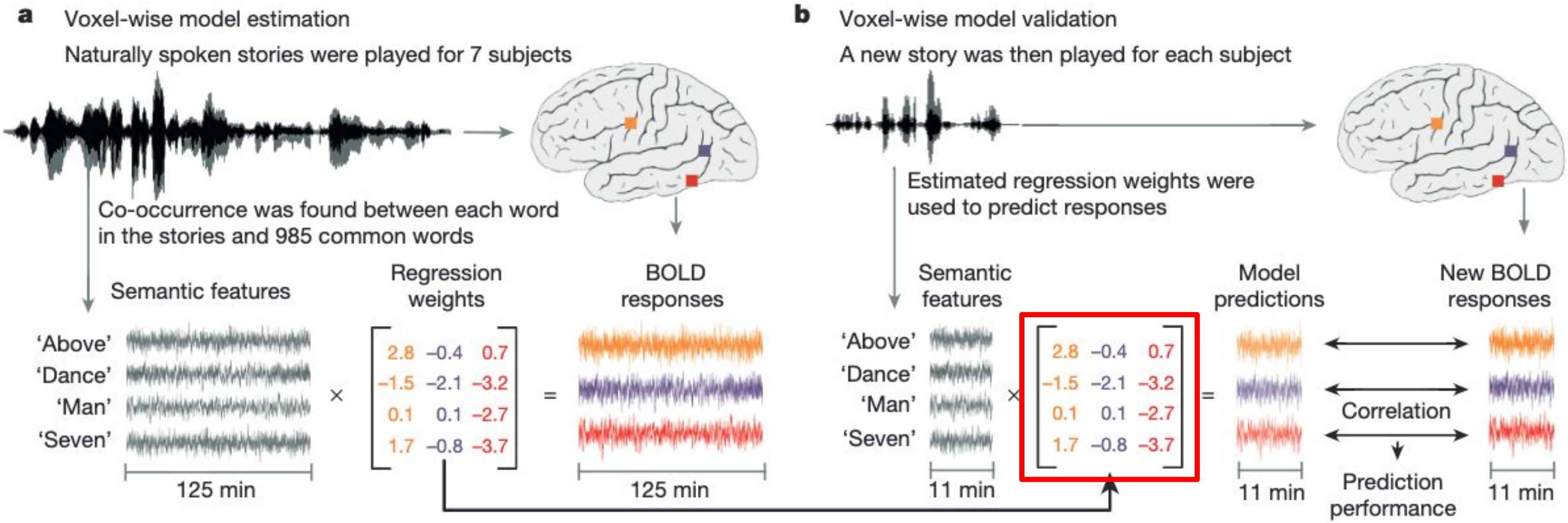
Model selection
(Training set)

Model assessment
(Test set)

Example: Huth et al., 2016



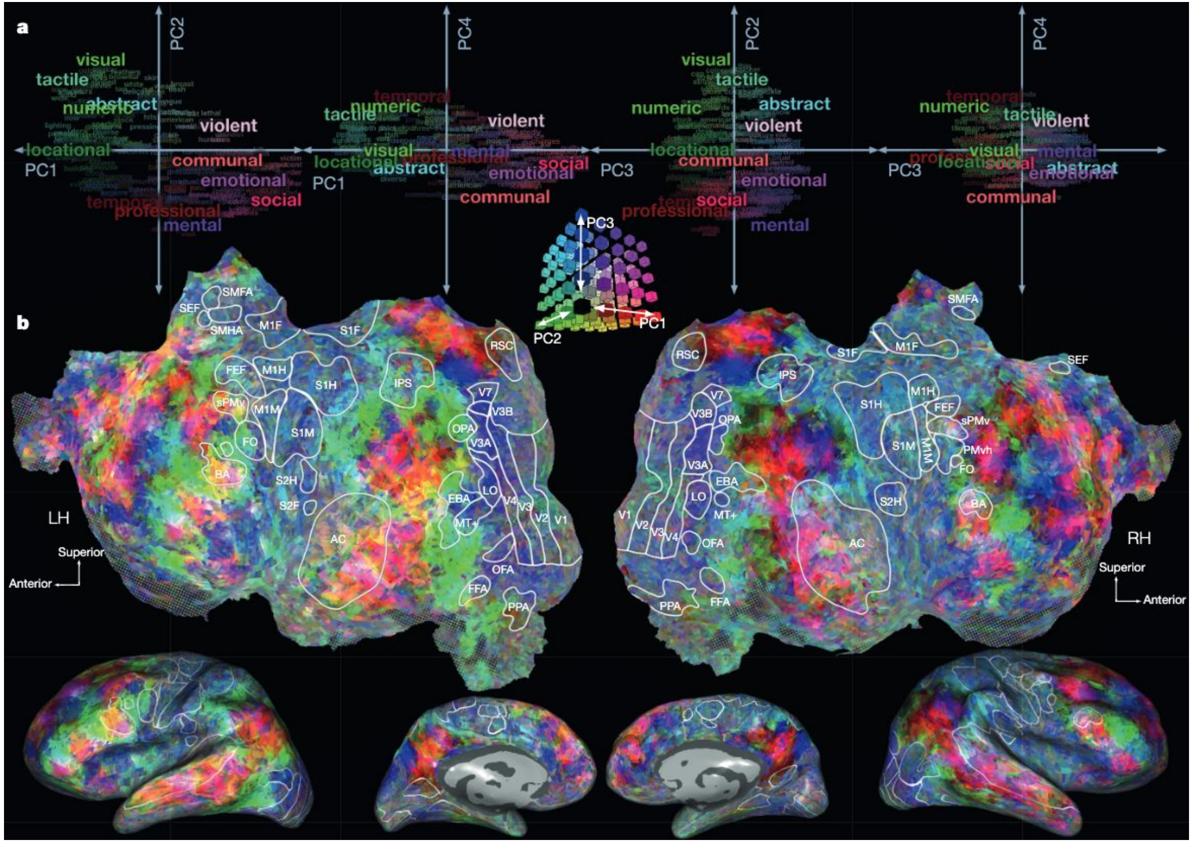
Example: Huth et al., 2016



Model selection
(Training set)

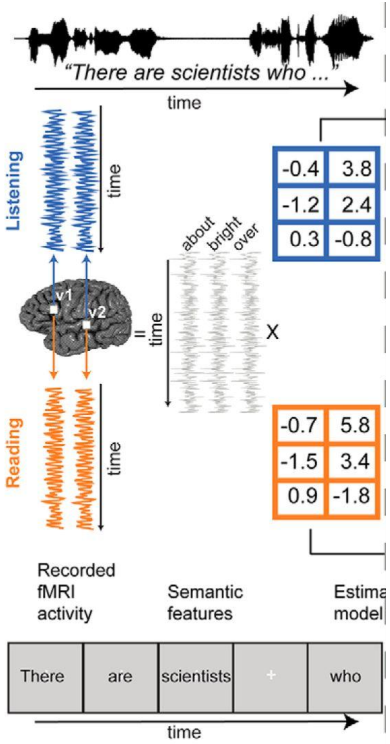
Model assessment
(Test set)

Example: Huth et al., 2016

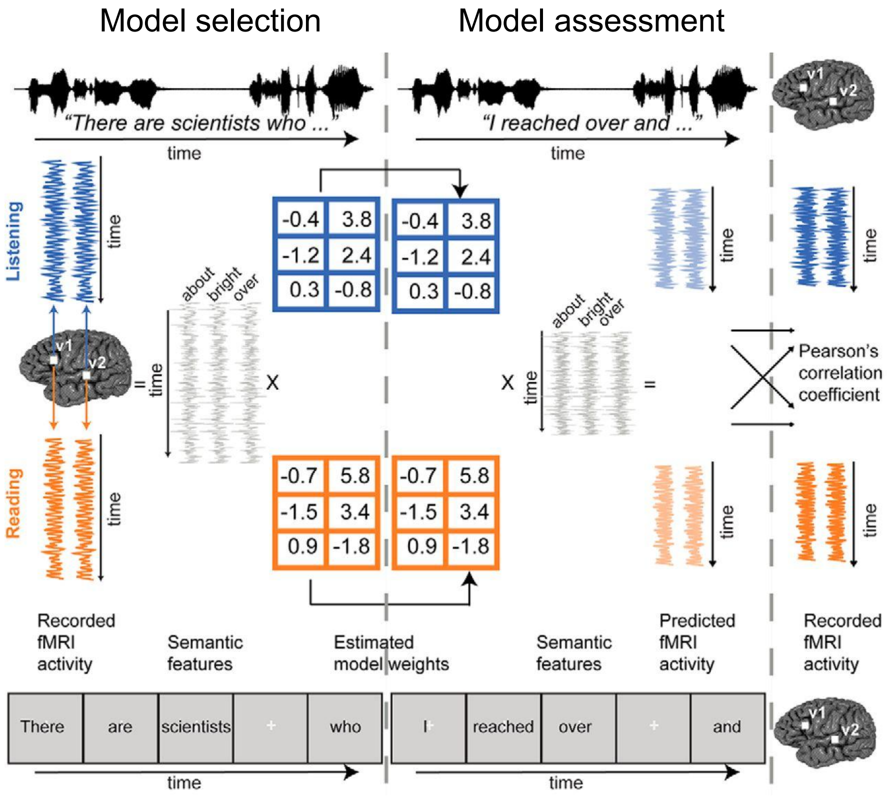


Example: Deniz et al., 2019

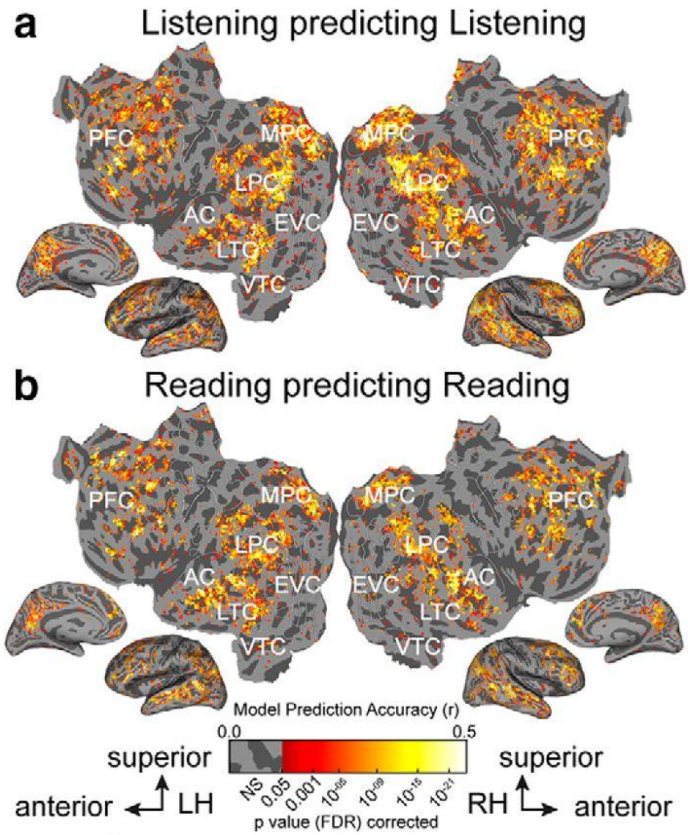
Model selection



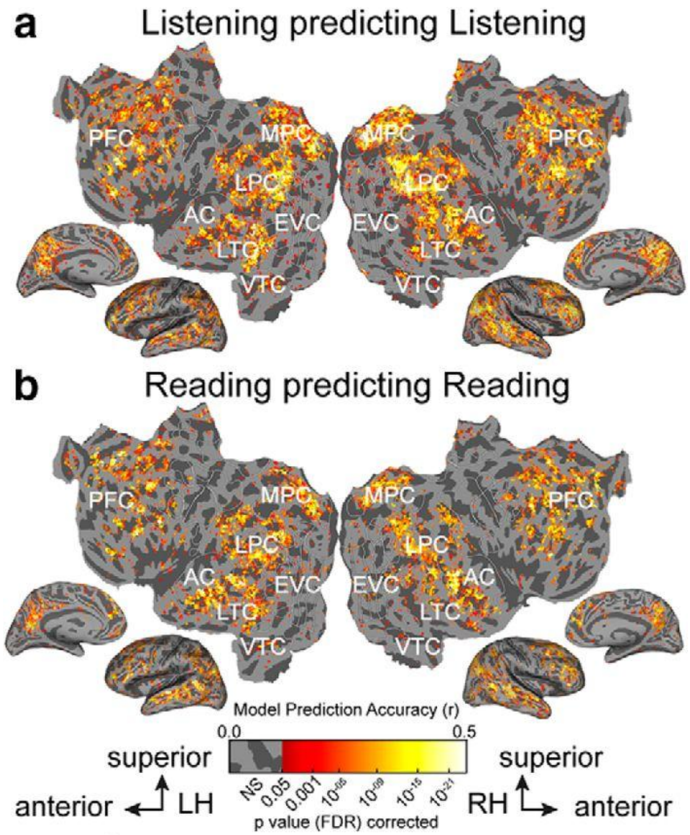
Example: Deniz et al., 2019

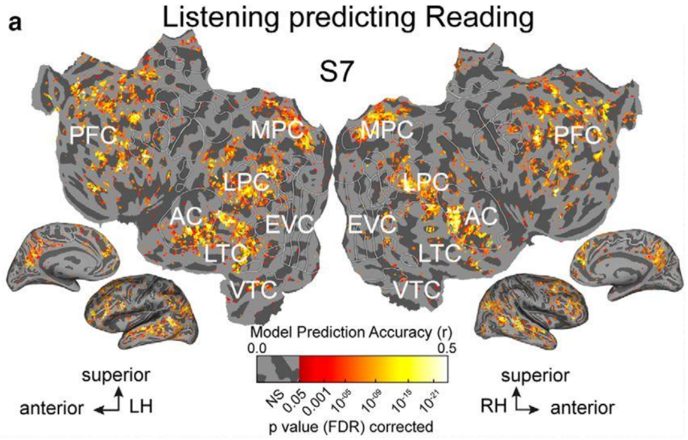


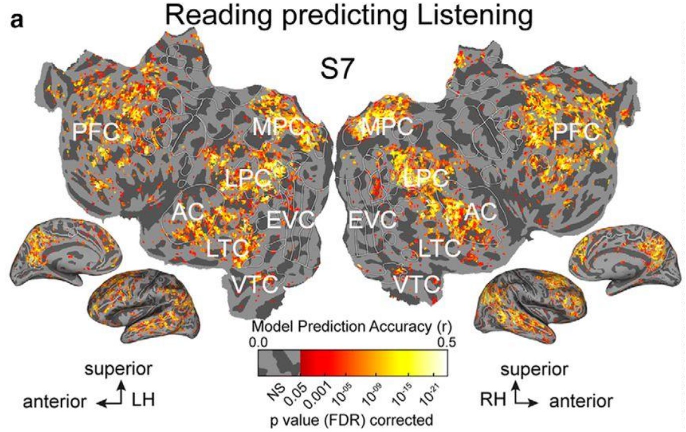
Example: Deniz et al., 2019



Example: Deniz et al., 2019





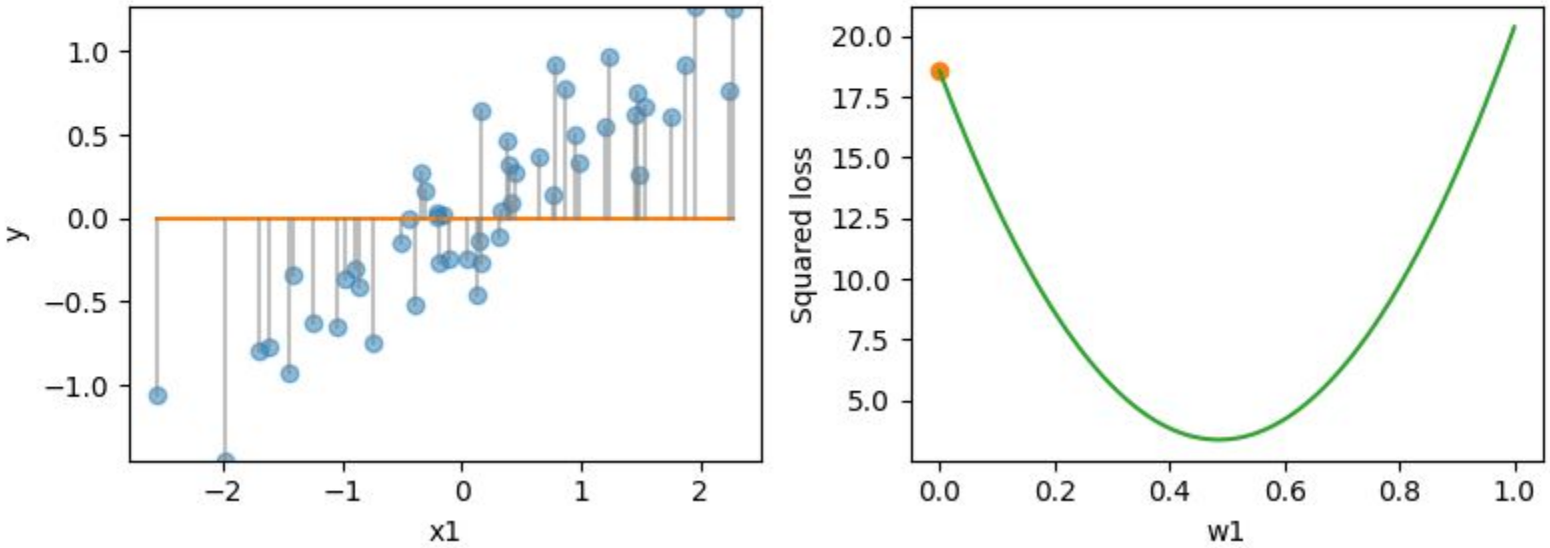


How to fit voxelwise models?

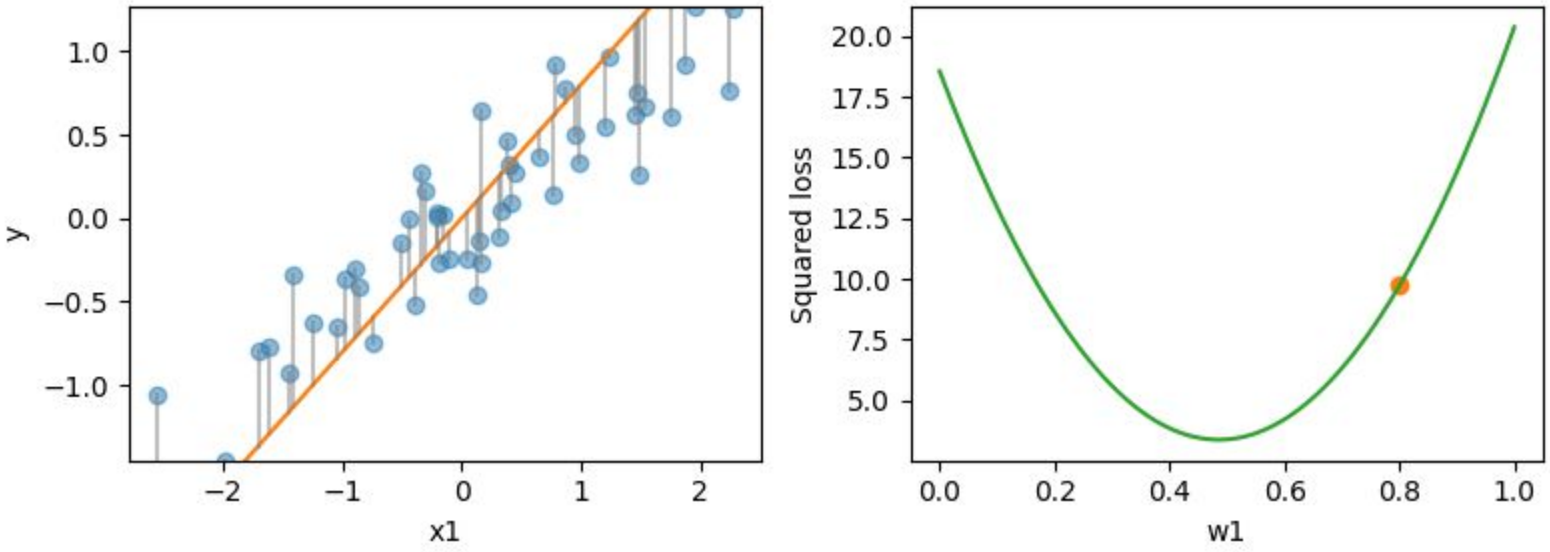
- Feature spaces describing the stimulus are high-dimensional
  - More dimensions than the number of samples available in the training set
- There is a high risk of overfitting: failure to generalize
- We need to use techniques from machine learning and data science to fit voxelwise models
  - Regularized regression
  - Cross-validation

Regularized linear regression

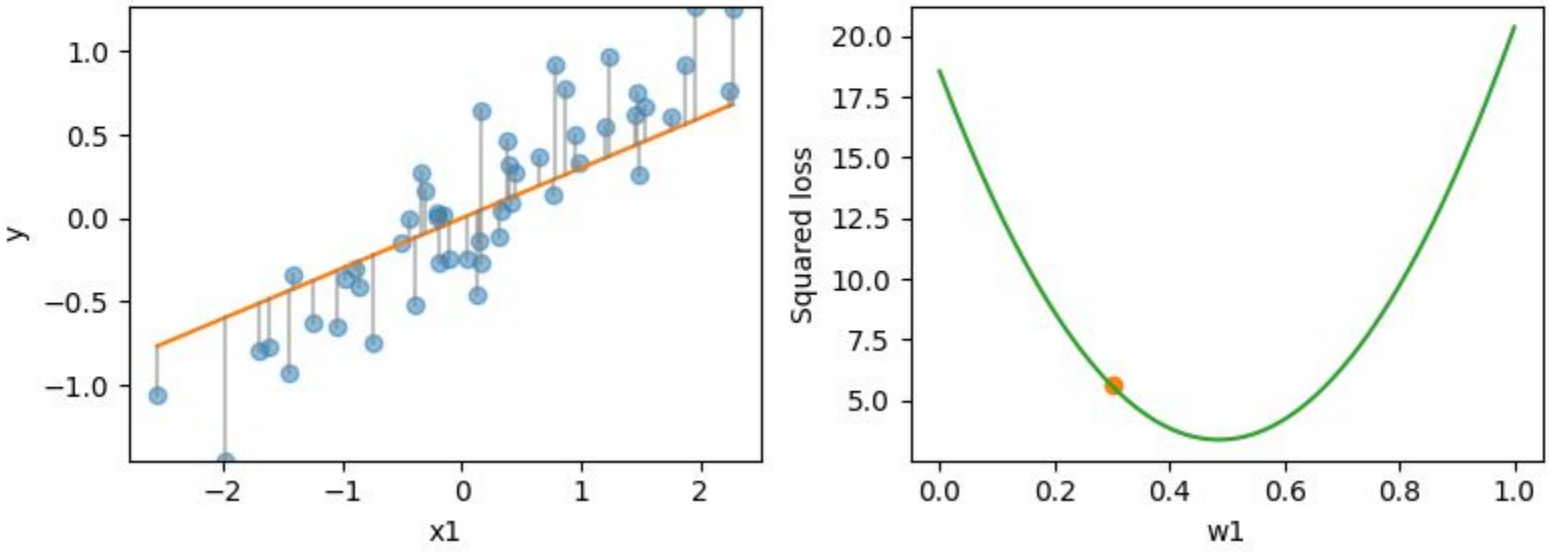
Linear regression



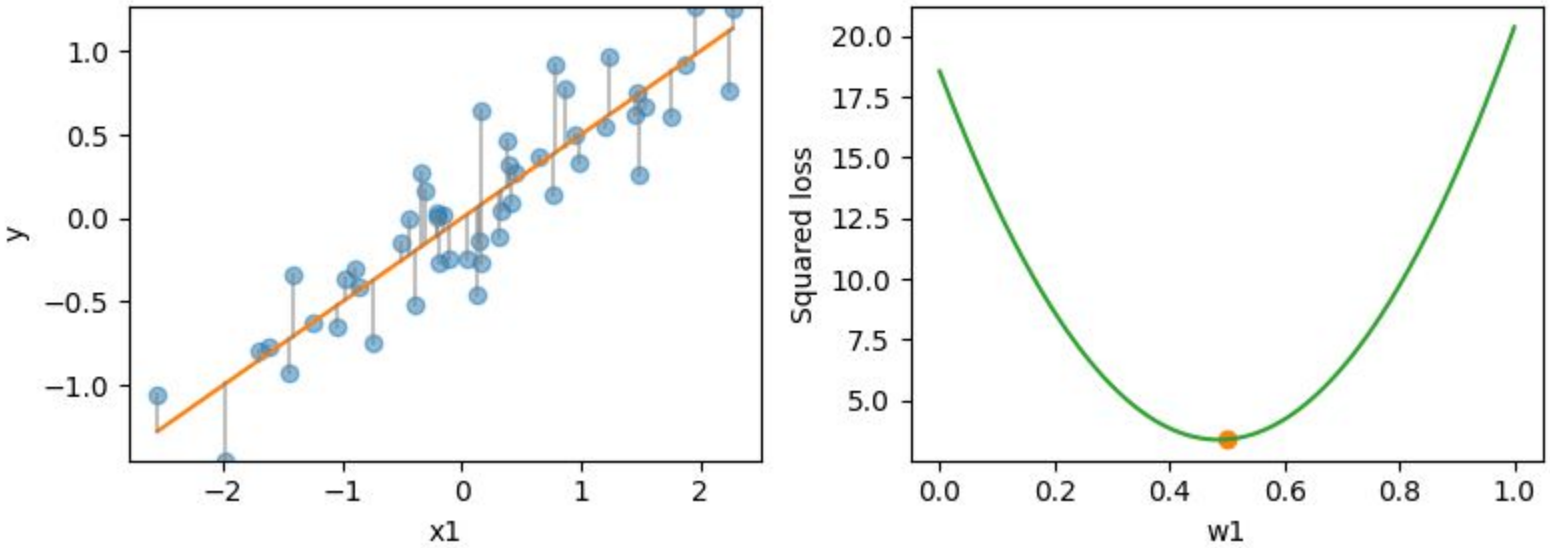
Linear regression



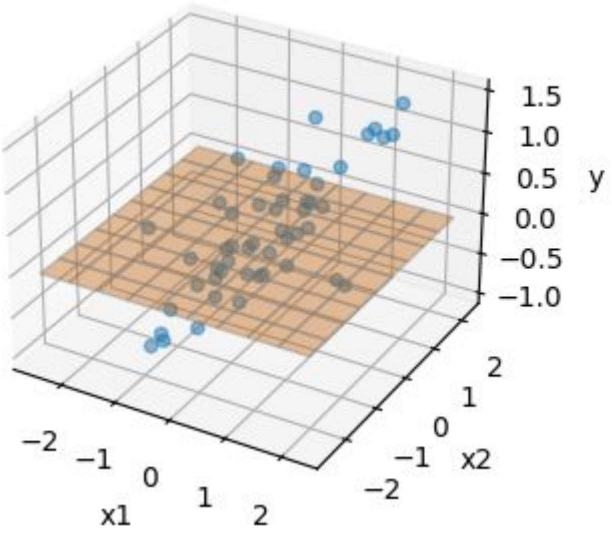
Linear regression

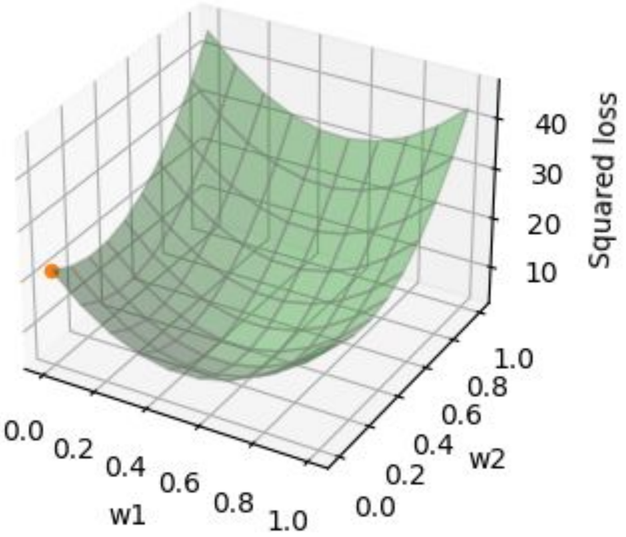


Linear regression

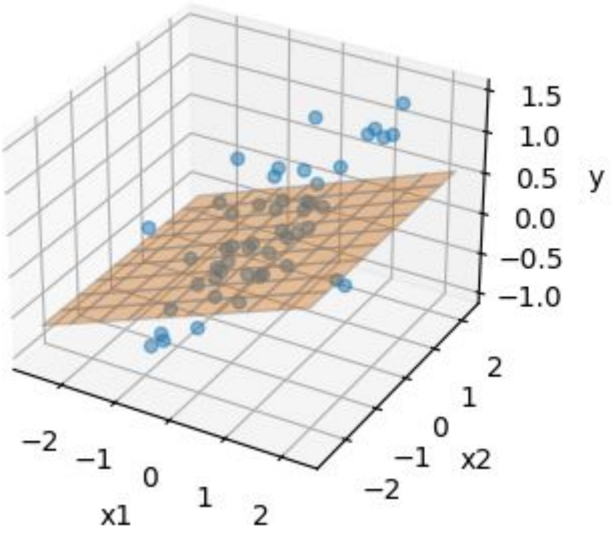


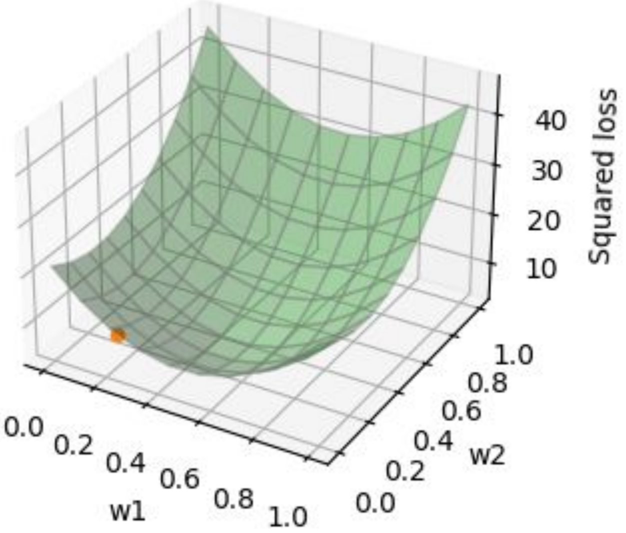
Multivariate linear regression



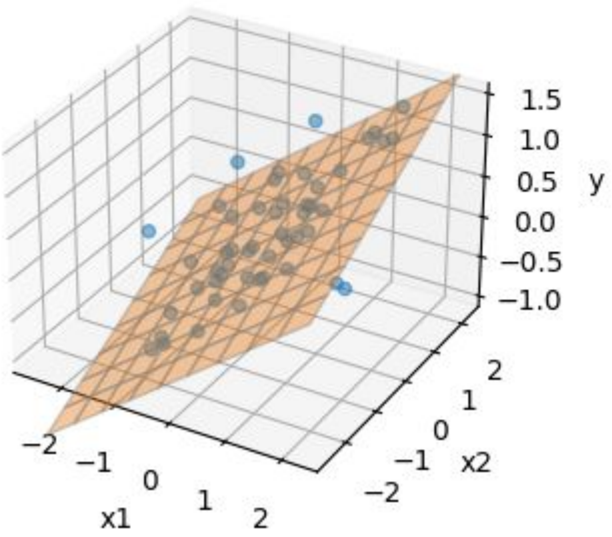


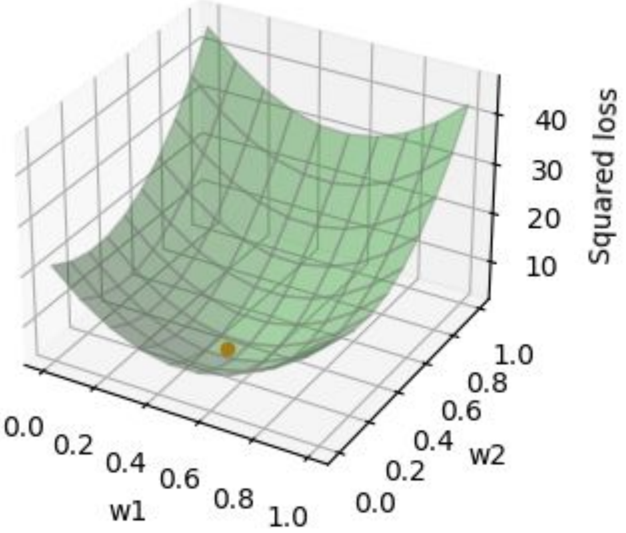
Multivariate linear regression



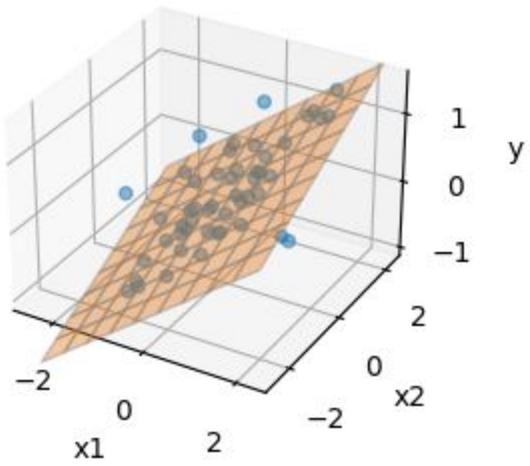


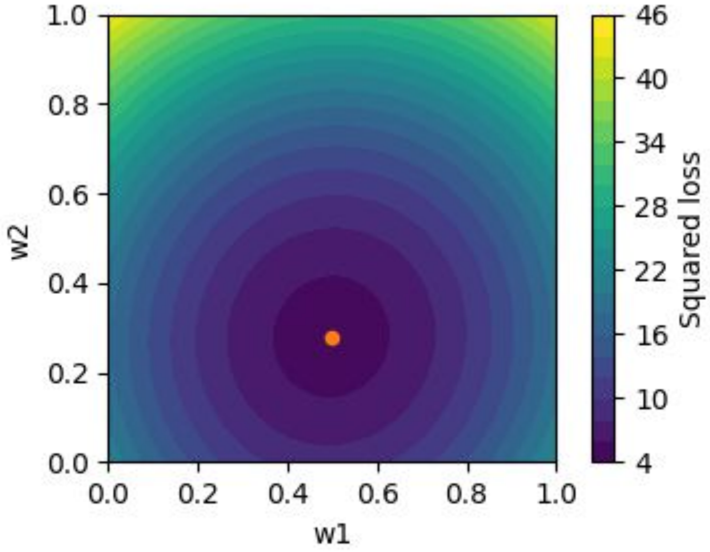
Multivariate linear regression



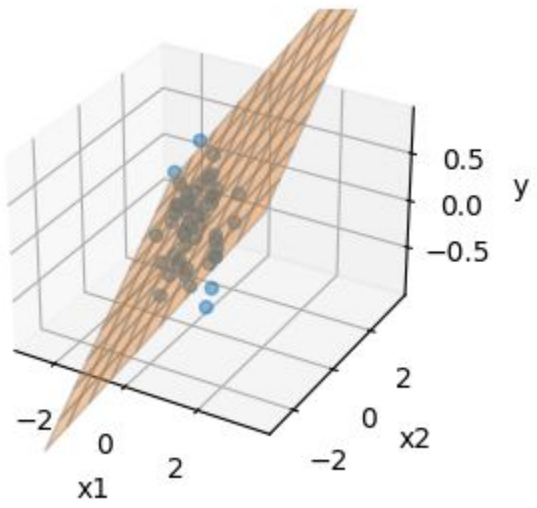


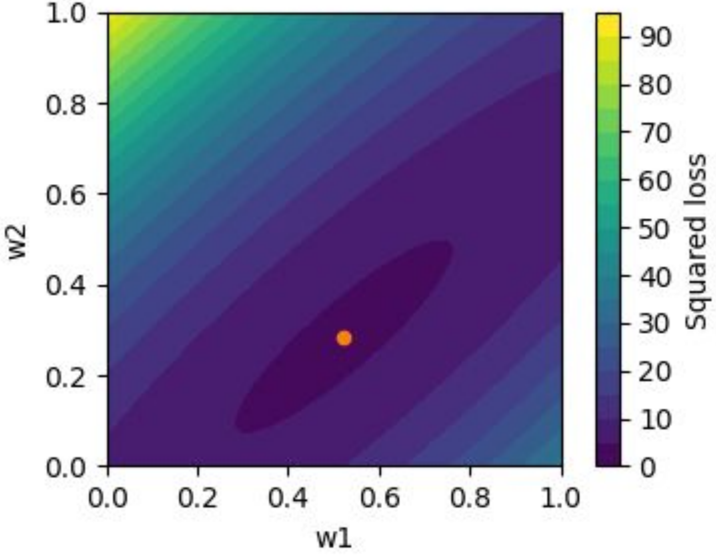
Multivariate linear regression



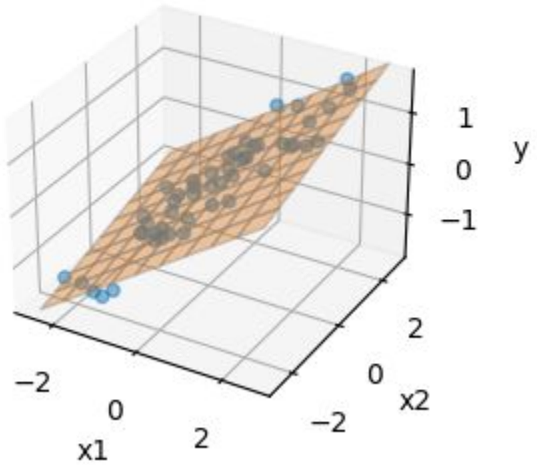


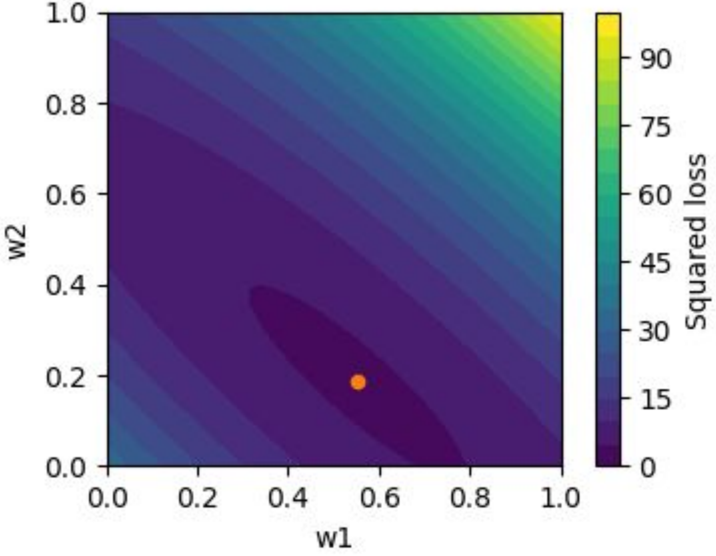
Multivariate linear regression - correlated features



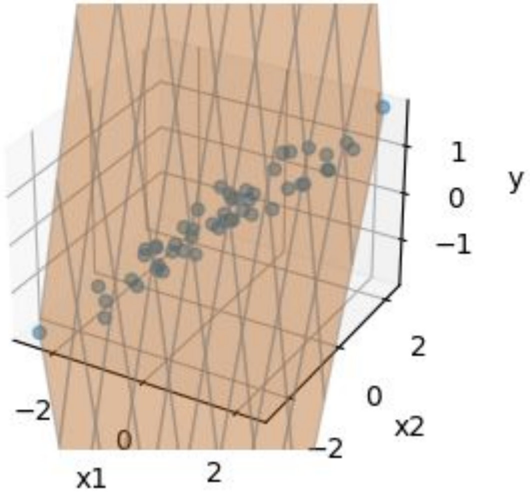


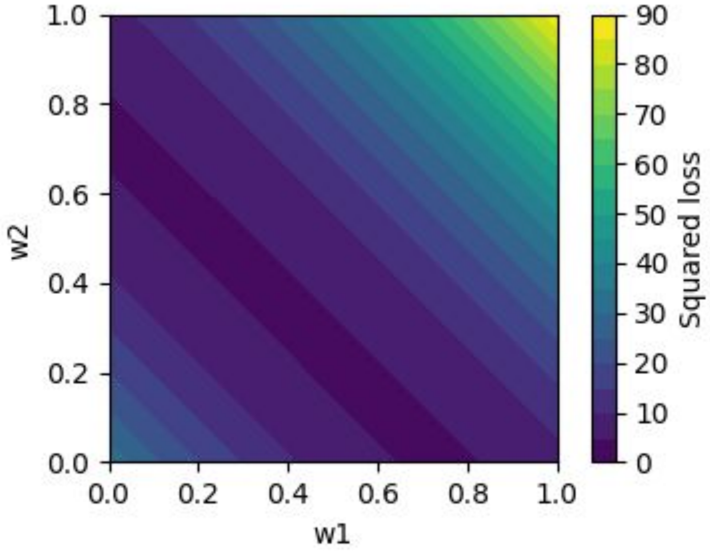
Multivariate linear regression - correlated features



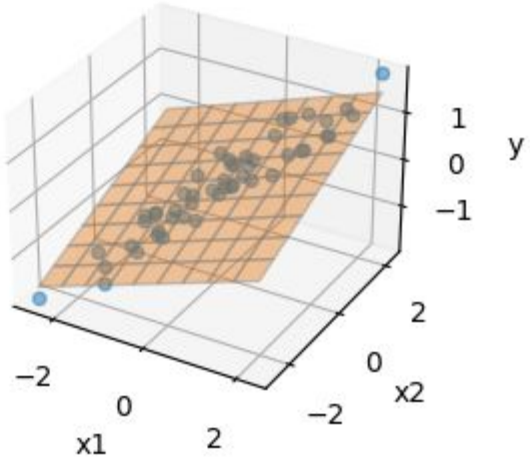


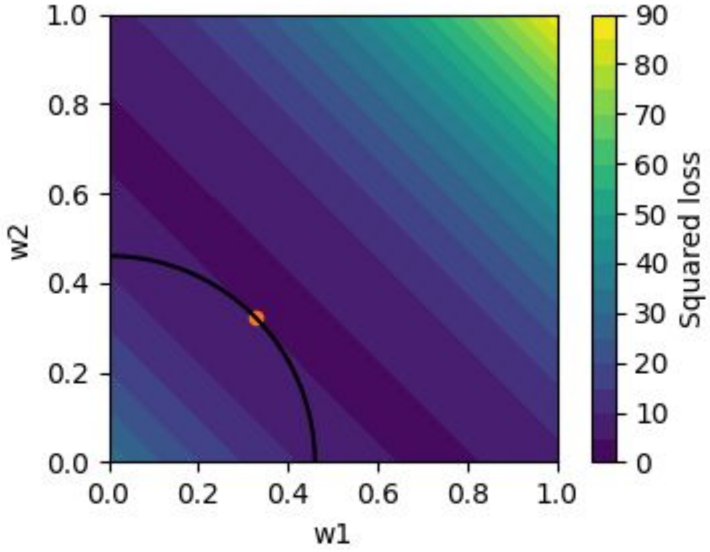
Multivariate linear regression - collinearity



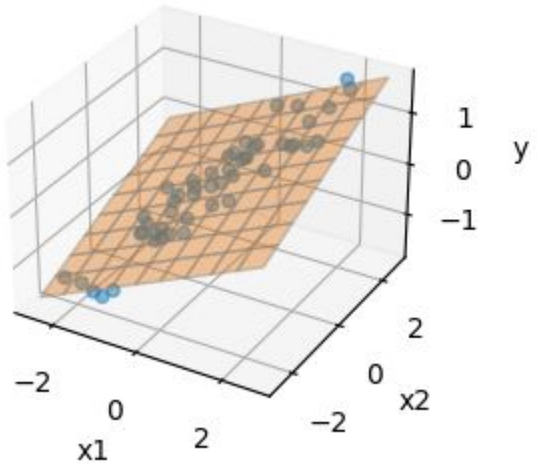


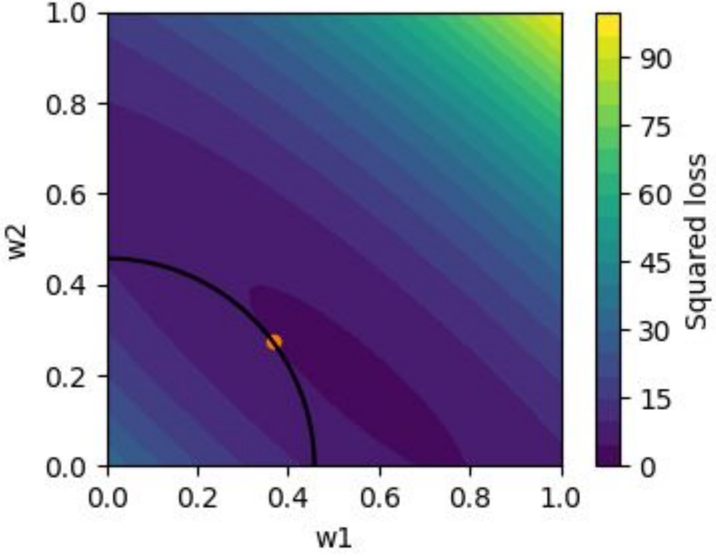
Multivariate linear regression - regularization (ridge)





Multivariate linear regression - regularization (ridge)





Ridge regression

Definition

Linear regression	$w^* = \text{argmin}_w \, \|\|y - Xw\|\|^2$
Ridge regression	$w^* = \text{argmin}_w \, \|\|y - Xw\|\|^2 + \alpha \, \|\|w\|\|^2$

Ridge regression

Definition

Linear regression $w^* = \text{argmin}_w \, ||y - Xw||^2$

Ridge regression $w^* = \text{argmin}_w \, ||y - Xw||^2 + \alpha \, ||w||^2$

Analytical solution

Linear regression $w^* = (X^T X)^{-1} \; X^T y$ λ_0^{-1}

Ridge regression $w^* = (X^T X + \alpha Id)^{-1} \; X^T y$ $(\lambda_0 + \alpha)^{-1}$

Ridge regression

Benefits

More robust with **correlated features**

Fix collinearity issues

Fix the case **n_features > n_samples** (underdetermined system)

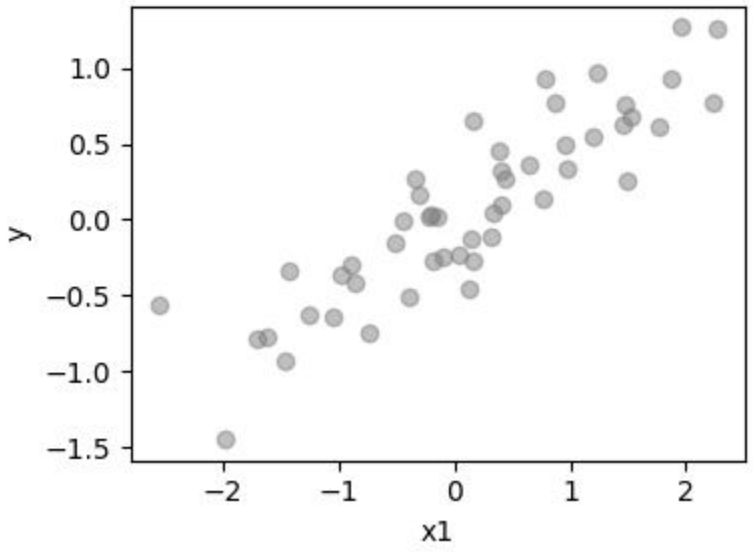
Drawback

Unknown hyperparameter α (theoretical link to the signal-to-noise ratio)

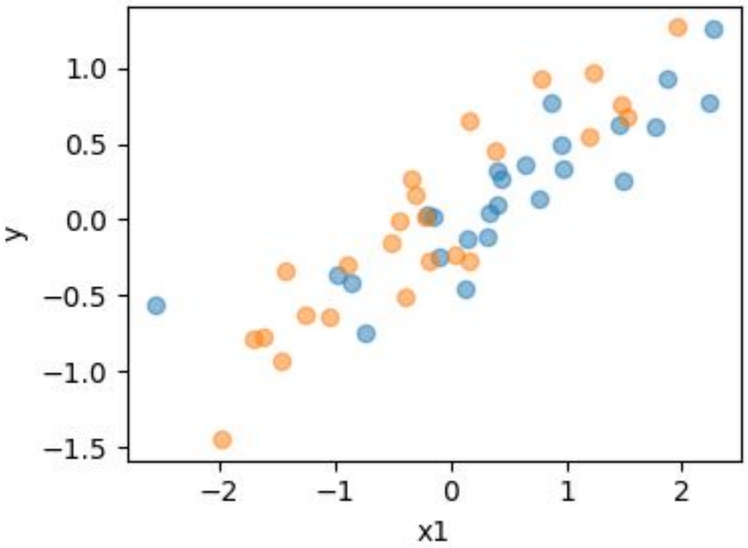
Solution

Cross-validation

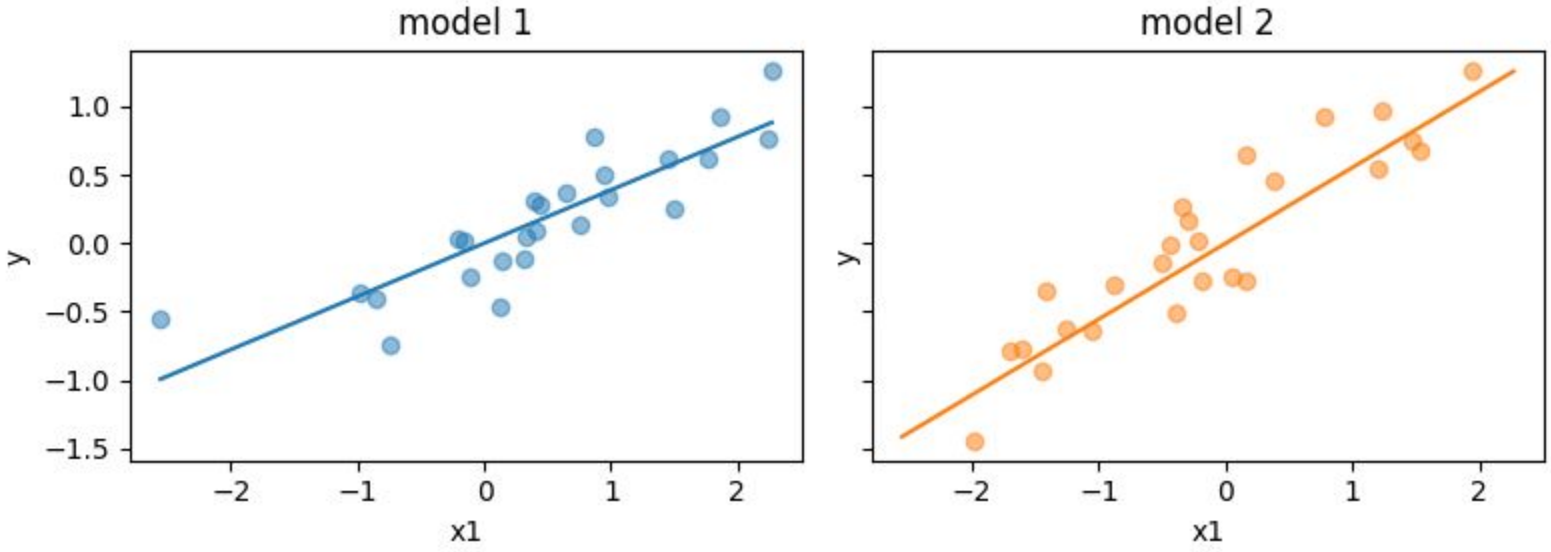
Cross-validation



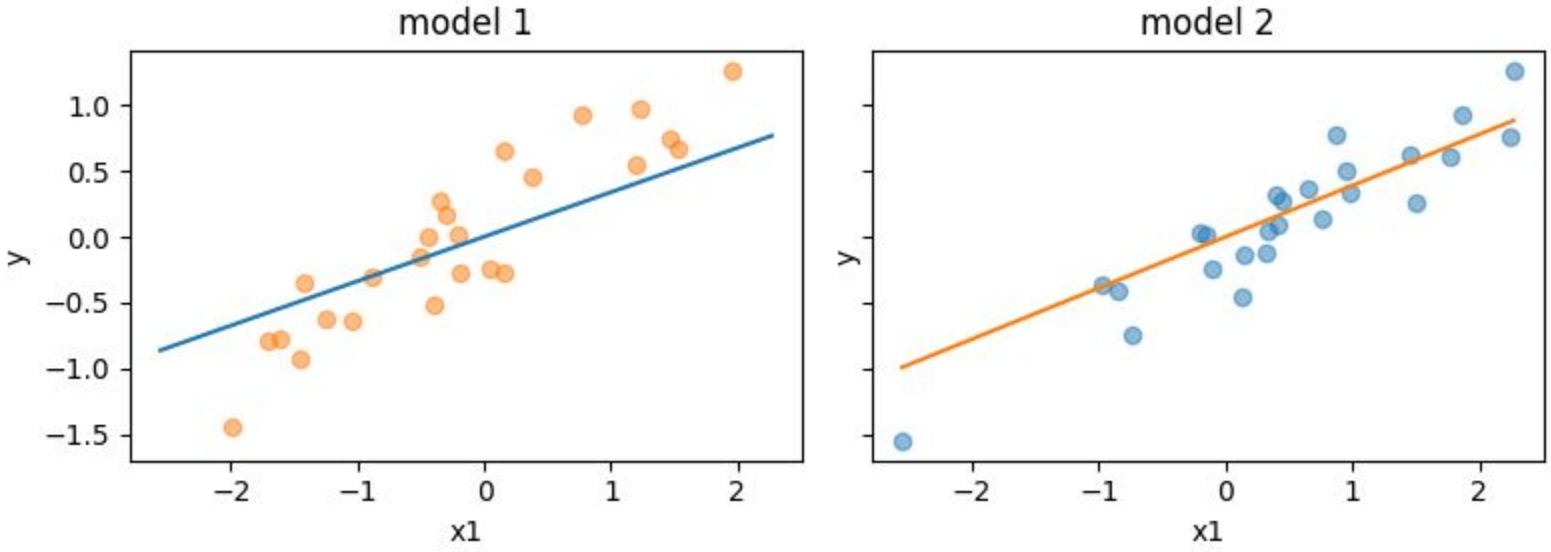
Cross-validation



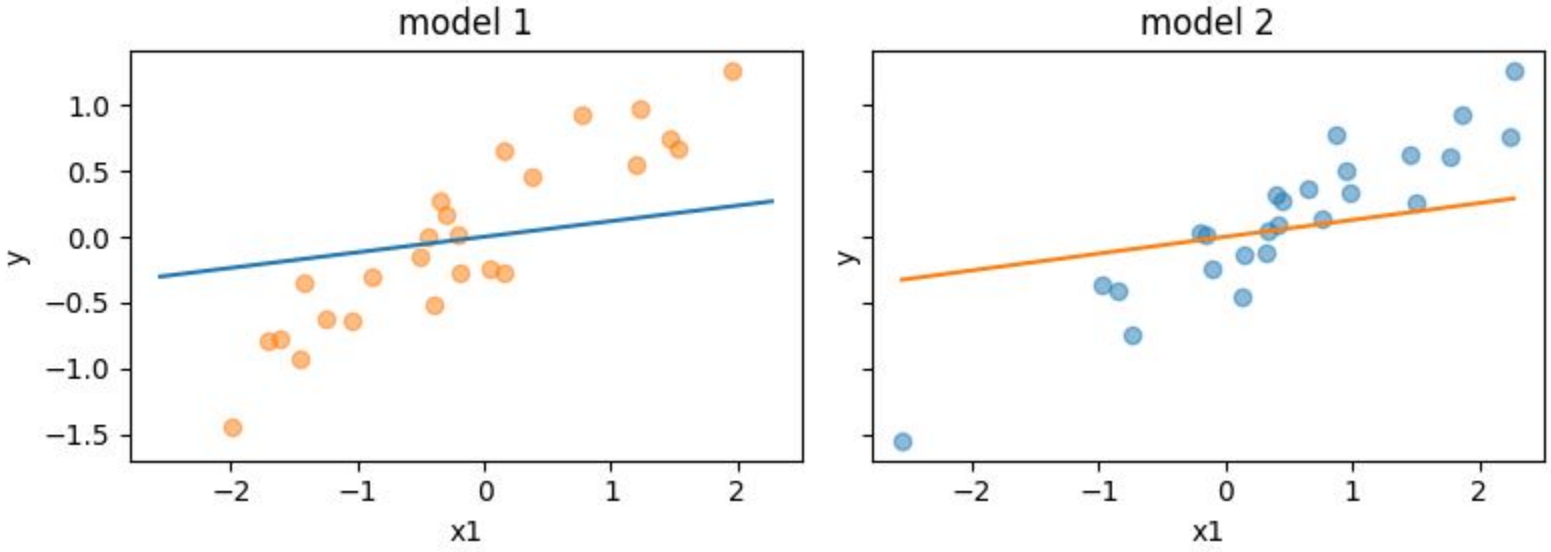
Cross-validation



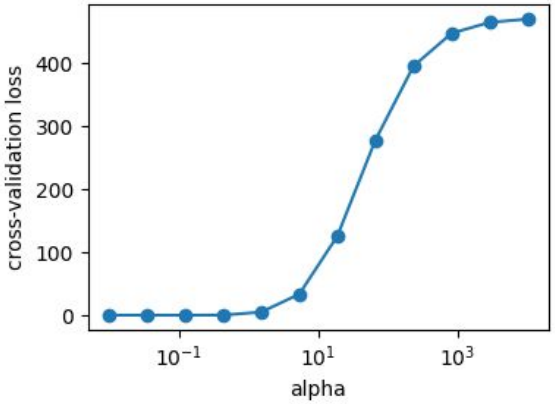
Cross-validation



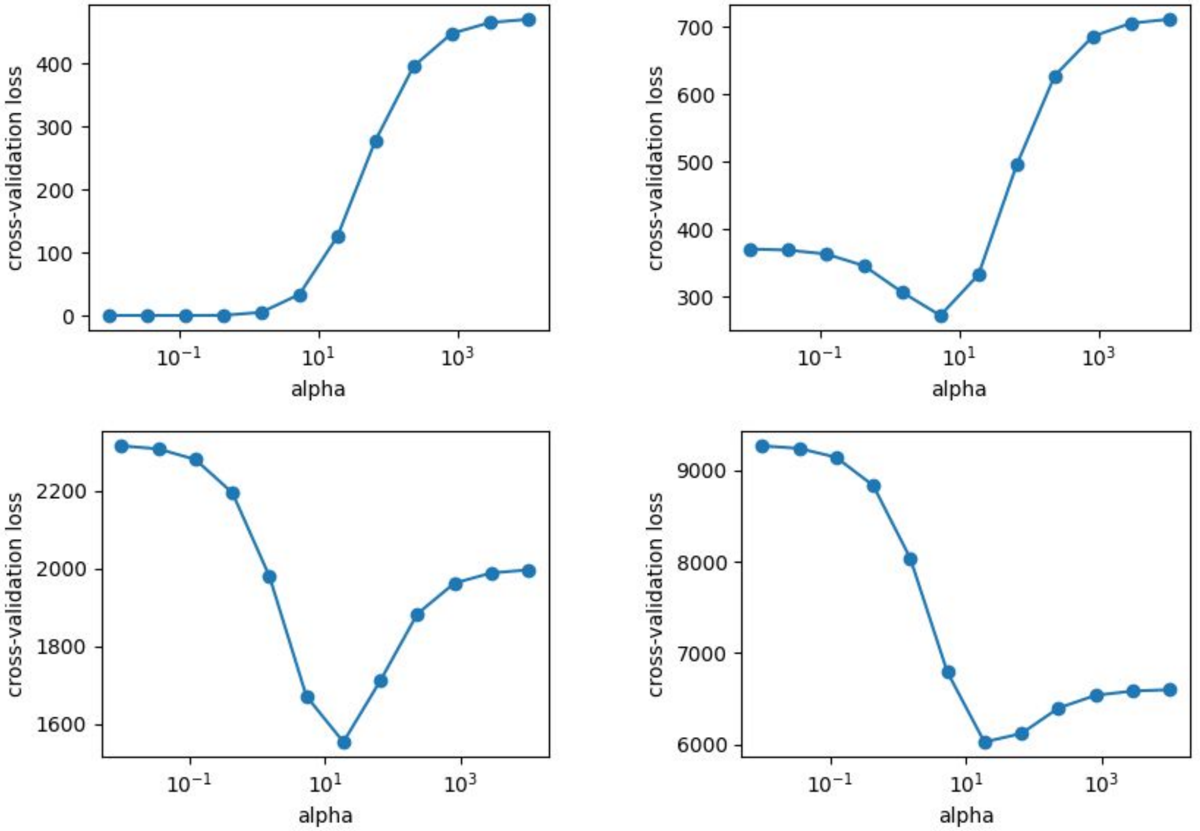
Cross-validation



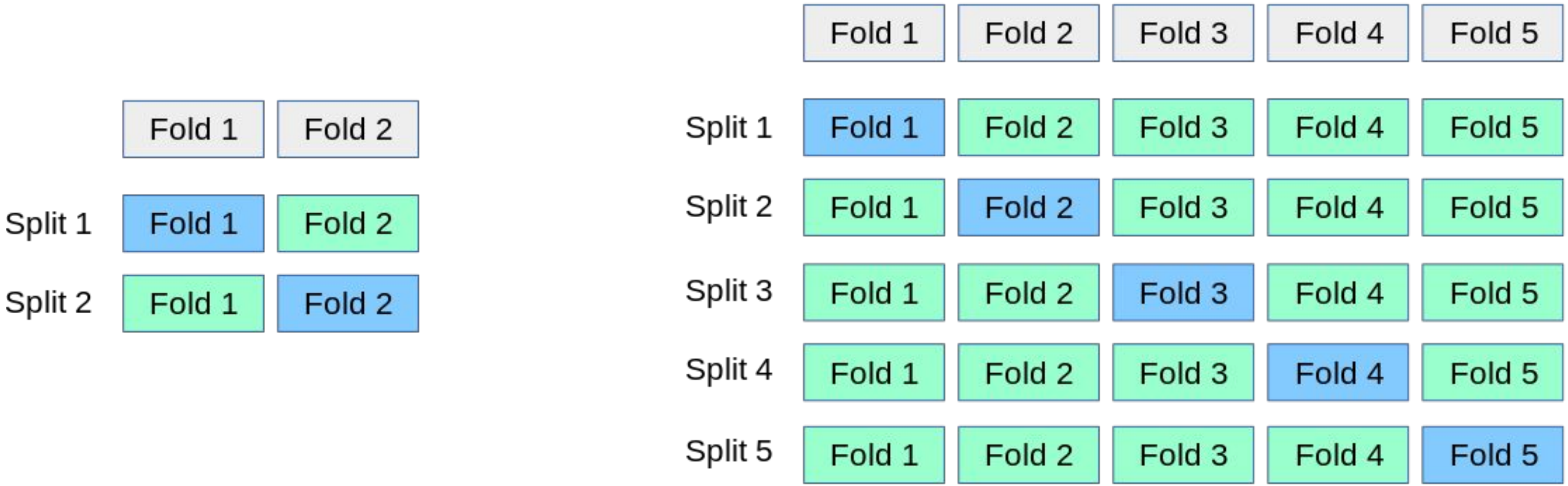
Hyperparameter path



Hyperparameter path



Cross-validation - more folds



Cross-validation - hyperparameter selection

for each **hyperparameter** candidate

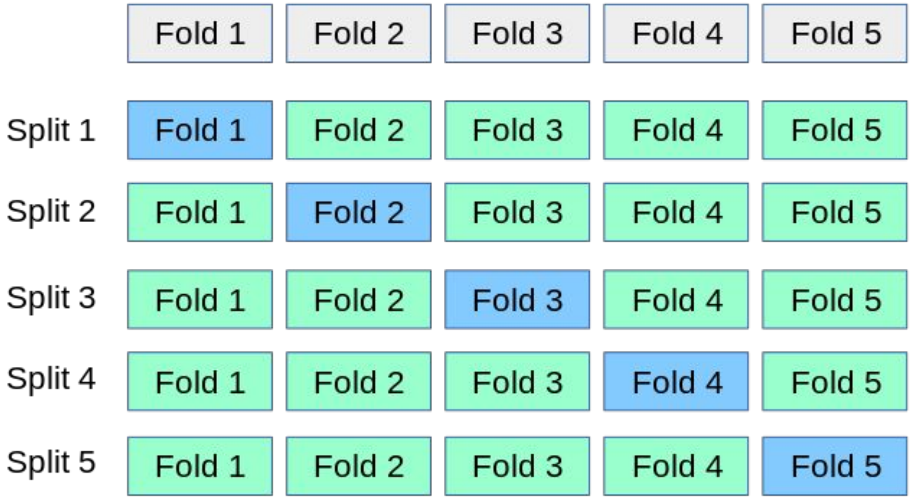
    for each split of the data

        fit a model on the training folds

        score the fitted model on the validation fold

    average scores over all splits

select best **hyperparameter**



Example

    Selection of α in ridge regression

Cross-validation - model selection

for each **model** candidate

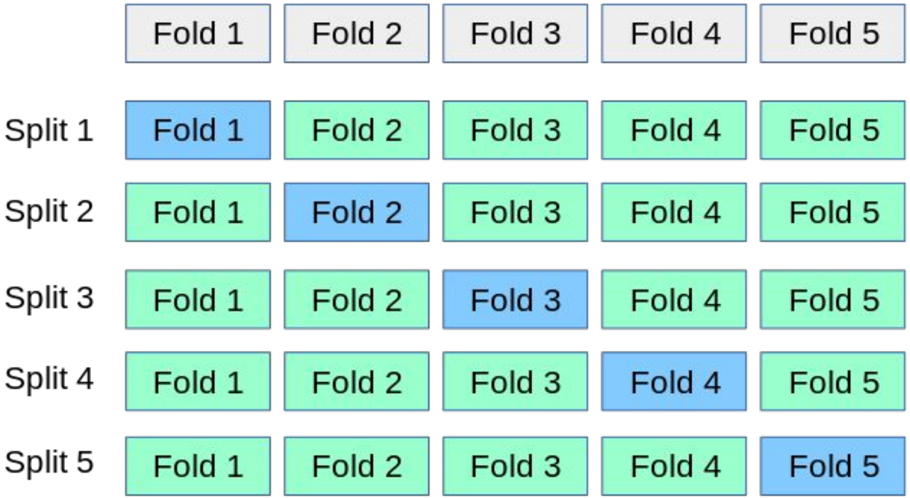
  for each split of the data

    fit a model on the training folds

    score the fitted model on the validation fold

  average scores over all splits

select best **model**



Example

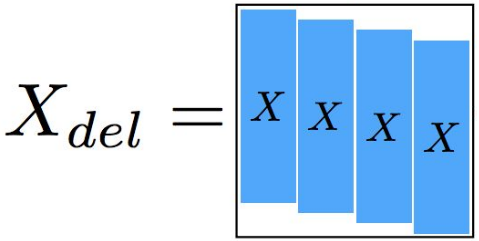
  Ridge regression versus Lasso

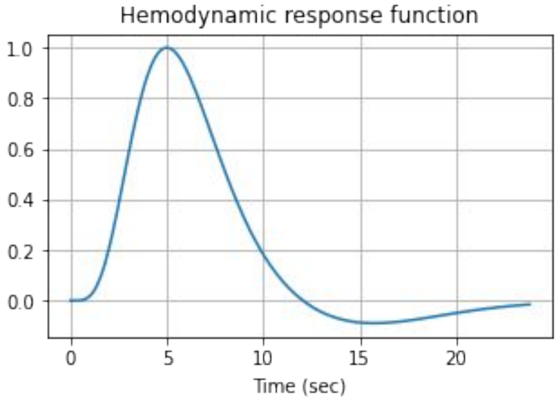
Model selection example - Time delays

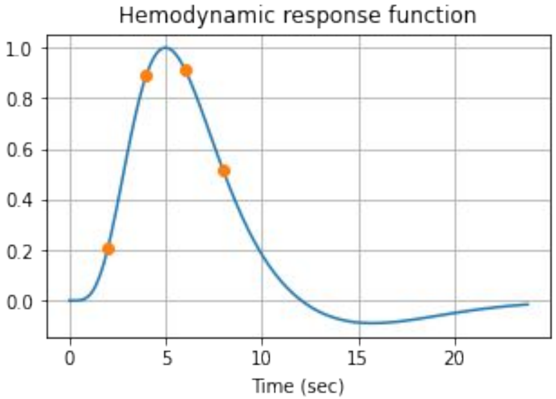
To model the **hemodynamic response function**

we copy all the features with different time delays

but how many delays is optimal ?







Model selection example - Time delays

To model the **hemodynamic response function**

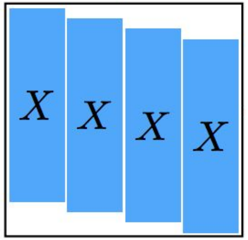
we copy all the features with different time delays

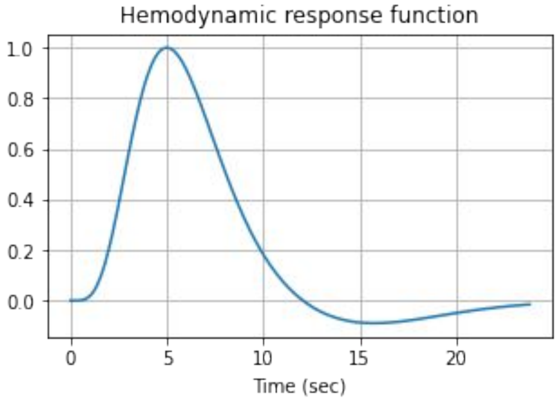
but how many delays is optimal ?

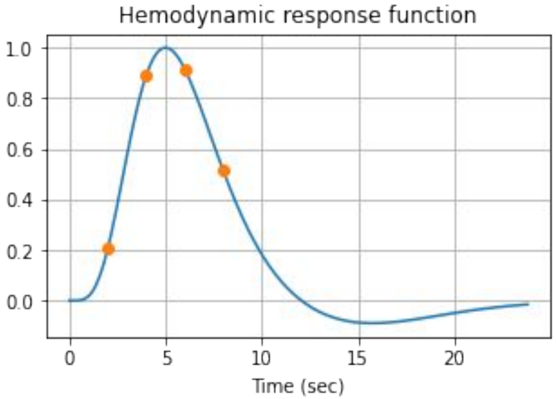
Method: cross-validation

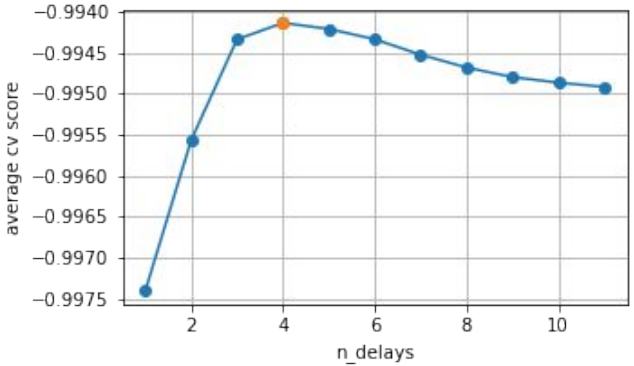
Answer: 4 (for this dataset)

$$X_{del} =$$

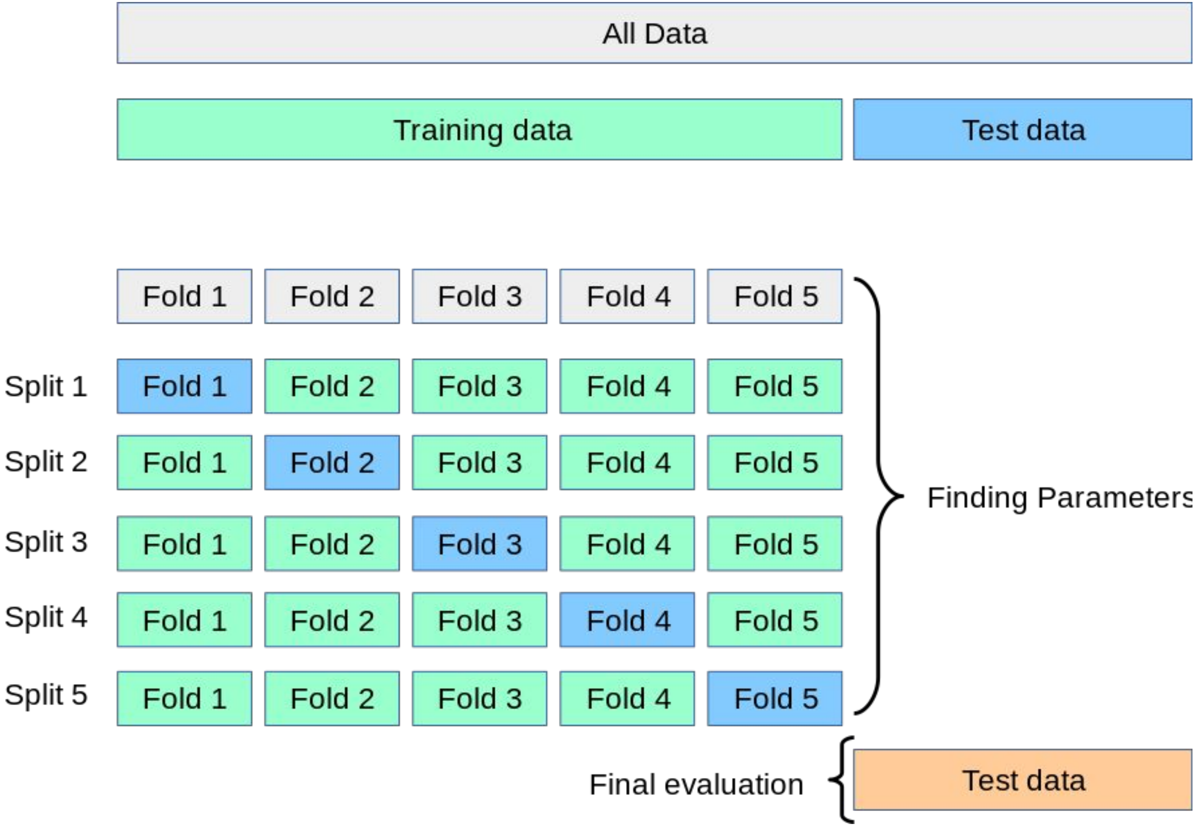








Generalization to new data



Generalization to new data

Generalization power

Estimated with prediction on a held-out test dataset

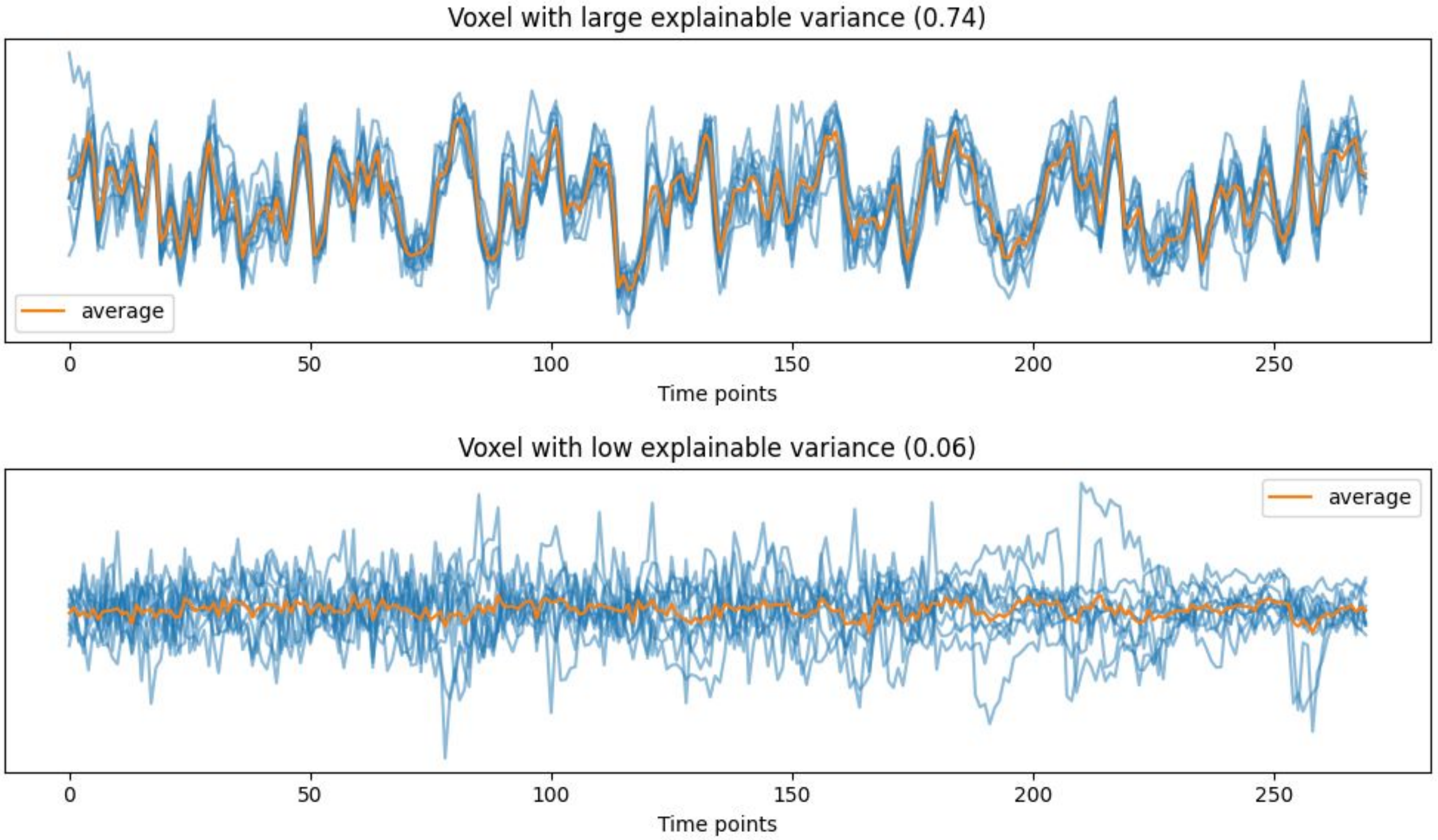
Generalization lower-bound (i.e. significance)

Estimated with permutations

Generalization upper-bound (i.e. explain**able** variance)

Estimated with repeats of the same stimulus

Explainable variance



Tutorials

https://github.com/gallantlab/voxelwise_tutorials

tutorials in python, notebooks style

voxelwise modeling helper functions

https://github.com/gallantlab/himalaya

python package, scikit-learn API, CPU/GPU

ridge-regression-like models for large number of voxels

(both repositories are still private for now)

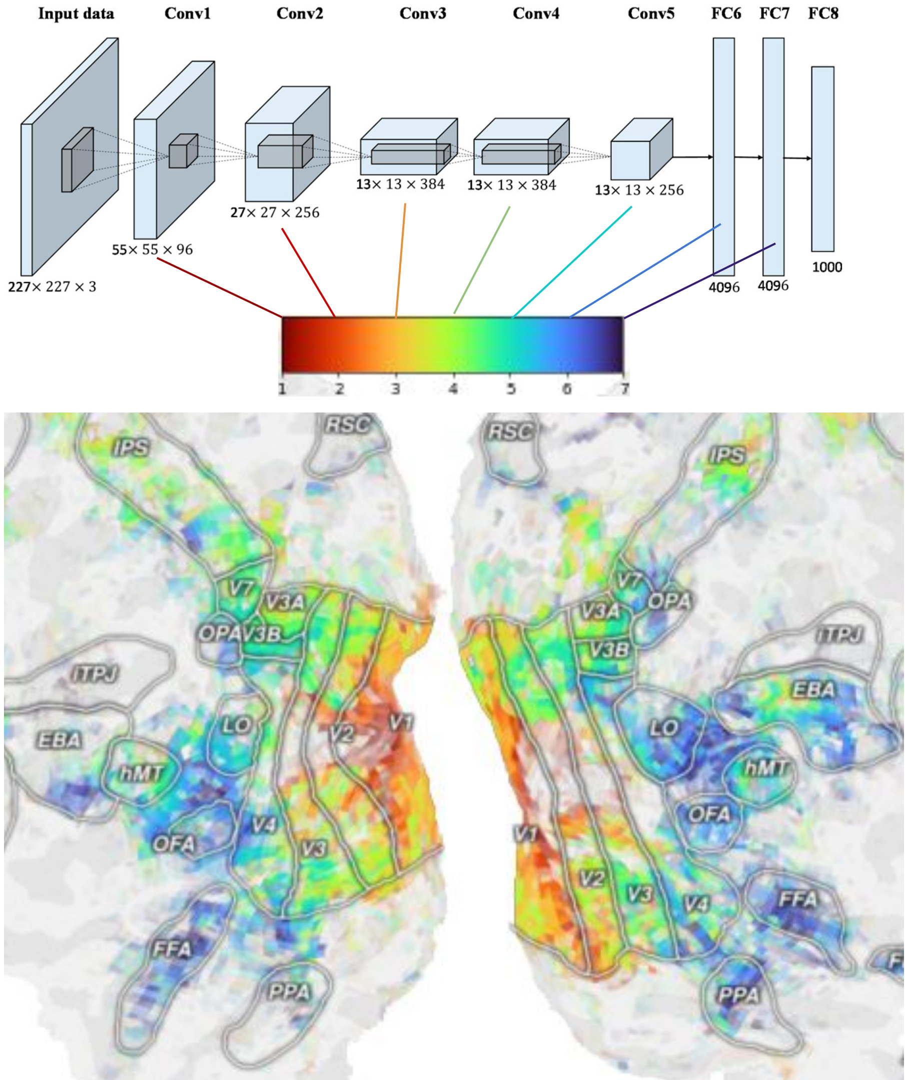
send me an email if you want an early access tomdlt@berkeley.edu

feedback much appreciated !

Advanced Voxelwise Modeling

Advanced use of the framework include:

- use very large number of features extracted from deep neural networks
- partition the explained variance over multiple feature spaces (with banded ridge regression)
- separate features over different timescales
- ...



Tutorials

(Fit a ridge model with wordnet features)

Association is not prediction

[*Statistical Modeling: The Two Cultures*, Breiman, 2001, Statistical science]
[*Statistics versus machine learning*, Bzdok et al., 2018, Nature Method]

"In the unfolding era of big data in medicine, the phrase "association is not prediction" should become as important as "correlation is not causation"."
[Bzdok et al., 2021, JAMA Psychiatry]

1 - Voxelwise modeling vs classical fMRI analysis

Comparison

- Classical: Block design, linear regression, t-test
- VM: Feature extraction, still a linear regression (!), but test set predictions
- Main difference: association/inference vs prediction - (old debate)

(inference = interpretable) vs (prediction = black box) ?

- no, we can still use linear models (!= random forest or neural networks)

Prediction is about replicability, generalization to new settings

- association can be highly dependent to particular subjects, cross-val less

Prediction estimates the effect size (explained variance)

- large significance (e.g. with many subjects) != large effect

Test set predictions largely reduces overfitting

- with enough features, one can explains 100% variance within set
  - even with linear models

2 - Voxelwise modeling

Regularized regression
- Reduces collinearity overfitting
- Reduces n_features > n_samples overfitting
- Handles different SNR per voxel

Model selection with cross-validation
- hyperparameter selection - example of ridge regularization
- model selection - example of the number of delays

Test set generalization as a final score
- generalization lower bound (ie significance) with shuffling
- generalization upper-bound (ie explainable variance) with repeats

Interpreting feature weights
- feature importance
- PCA

Tutorials