Improving brain decoding interpretability using structured sparsity methods

José P Valdés-Herrera

Retreat Braunlage, 2018

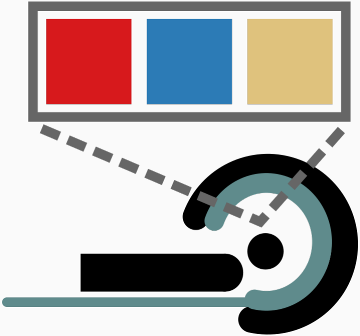
DZNE
Deutsches Zentrum für Neurodegenerative Erkrankungen
in der Helmholtz-Gemeinschaft

Aging & Cognition
Research Group

fMRI refresher

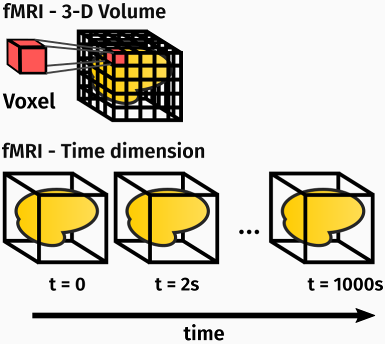
BOLD fMRI

My work mainly focuses on analysis of task-related fMRI data. During scanning, the participant performs a behavioral task.



fMRI data

The data resulting from fMRI is 4 dimensional (structural is 3-D). The extra time dimension allows us to analyse how brain activity changes during the task.

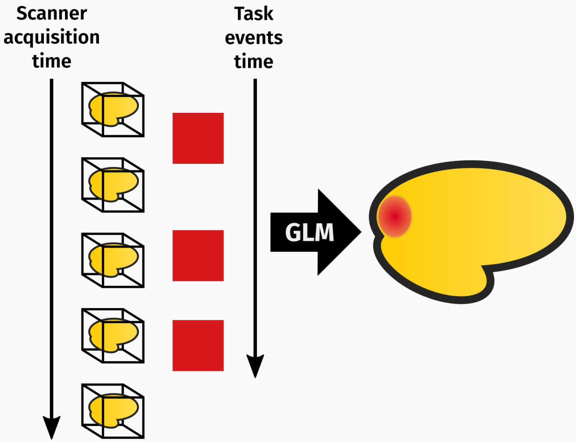


A typical fMRI session consists of:

- 600-900 3-D volumes with 10^5 to 10^6 voxels each.
- Voxel resolution varies from 1 (ultra-high) to 3 mm^3.

Putting it all together

We can analyse the data to detect changes in activation using a General Linear Model (GLM), a massive-univariate analysis.

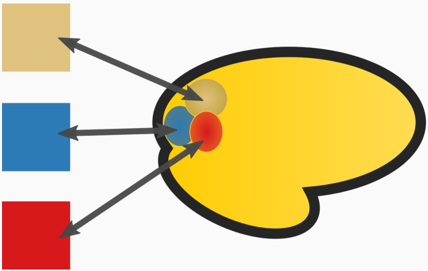


Decoding

Asking about categories

Maybe we would like to ask different scientific questions:

1. Can we distinguish categories from the task?
2. Can we find cluster of voxels that discriminate among those categories?



Model

Our proposal to answer those questions is to use a linear model.

$$y = Xw$$

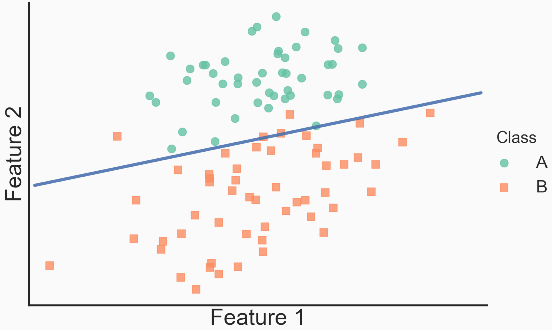
But, we have too many voxels (features) and too few images (samples). That is why we add an extra term to the problem to be solved called regularization, $J(w)$.

$$\hat{w} = \operatorname{arg\,min}_w \mathscr{L}\left(y, X, w\right) + J(w)$$

The solution that we obtain is a vector of weights, $\hat{w}$ with one weight per voxel: the weight map.

If things were this easy...

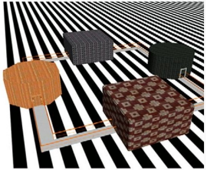
Only two features (e.g., voxels) and many samples:

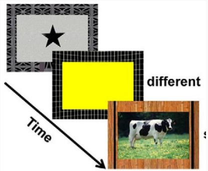


In neuroimaging, we have thousands of features and we cannot plot the data. Instead, we show the weight map.

A real experiment

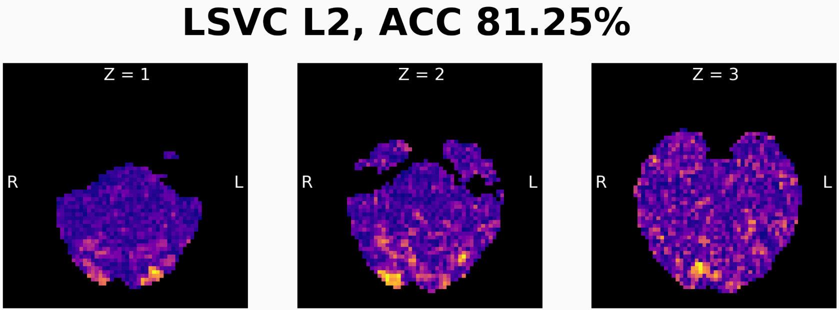
In this task, participants could be in any of four different rooms[1].





[1] Shine, J. P. et al. *Journal of Neuroscience* **2016**, *36*, 6371–6381.

Decoding a real experiment



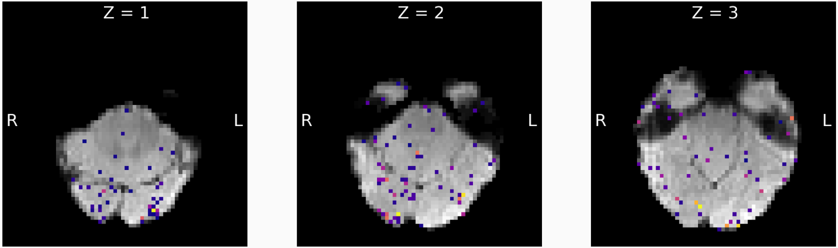
1. The accuracy (ACC) tells us most of the time the right room is identified (25% is random guess).
2. The weight map is *dense*: all weights are $\neq 0$.

Which voxels are the most relevant?

Sparsity

We can also find a *sparse* solution, i.e., many voxel weights are set to 0.

LSVC L1, ACC 84.77%



1. Accuracy is even better.
2. Non-relevant voxels are set to 0, but some relevant may be too (e.g., correlated voxels)!

Is the solution unique?

Structured sparsity

Why structured sparsity?[2]

Summary so far:

	Pros	Cons
Dense	stable	all voxels appear relevant
Sparse	few chosen voxels	unstable

[2] Baldassarre, L. et al. *2012 Second Int. Work. Pattern Recognit. NeuroImaging*.

Why structured sparsity?[2]

Summary so far:

	Pros	Cons
Dense	stable	all voxels appear relevant
Sparse	few chosen voxels	unstable

Structured sparsity offers a middle ground solution that is stable and selects whole relevant areas.

[2] Baldassarre, L. et al. *2012 Second Int. Work. Pattern Recognit. NeuroImaging*.

Structured sparsity

In a decoding analysis wishlist,

- take into consideration spatial and temporal information,

Structured sparsity

In a decoding analysis wishlist,

- take into consideration spatial and temporal information,
- recover the *true* weight maps,

Structured sparsity

In a decoding analysis wishlist,

- take into consideration spatial and temporal information,
- recover the *true* weight maps,
- ease interpretation of results.

Structured sparsity

In a decoding analysis wishlist,

- take into consideration spatial and temporal information,
- recover the *true* weight maps,
- ease interpretation of results.

To favour structured sparsity solutions, we need to use particular regularization terms.

BrainOwl is a classifier based on the *Ordered Weighted* l_1 (OWL)[3] norm.

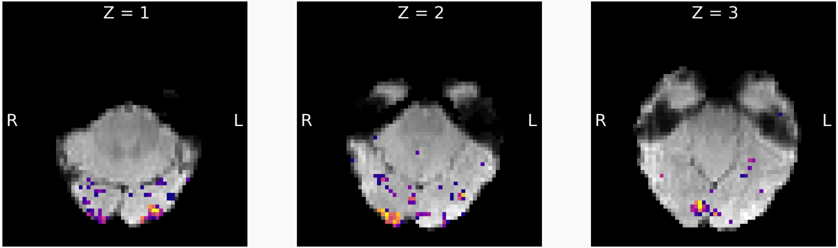
$$J_v(w) = \sum_{i=1}^{n} |w|_{[i]} v_i = v^T |w|_{\downarrow}$$

The OWL norm is robust to correlations and can be implemented efficiently.

[3] Zeng, X.; Figueiredo, M. A. T. *arXiv* **2015**, Bogdan, M. et al. *The Annals of Applied Statistics* **2015**.

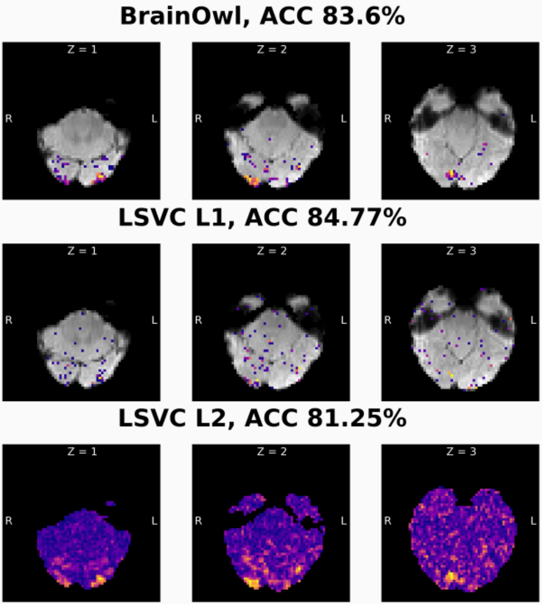
Rooms question revisited

BrainOwl, ACC 83.6%



1. High accuracy ensures we can distinguish rooms.
2. Weight map shows now only selected areas.

Summary: dense, sparse, and structured sparse



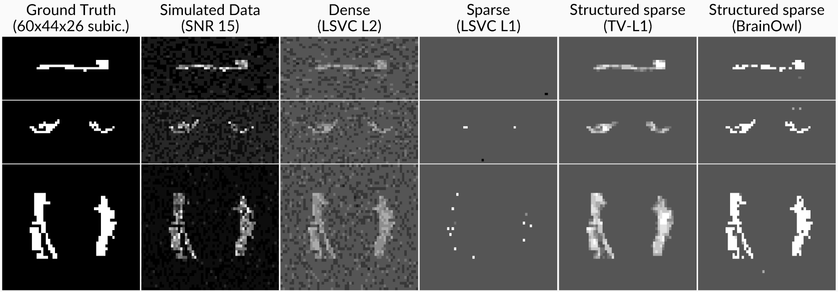
Software

- BrainOwl
  `github.com/jpvaldes/brainowl`
  Soon in `www.wolberslab.net`
- nilearn contains alternative decoders (TV-L_1, GraphNet)
  `nilearn.github.io`
- scikit-learn
  `scikit-learn.org`

Thank you!

Questions?

Other structured sparsity decoders



There are other structured sparsity classifiers implemented like *Sparse Total Variation* (TV-l_1)[4] or *Graph-Net*[5].

---

[4] Gramfort, A. et al. *2013 Int. Work. Pattern Recognit. Neuroimaging.*
[5] Grosenick, L. et al. *Neuroimage* **2013**, *72*, 304–321.

Structured Sparsity: Graph-Net[6]

Basic idea: look for a regularization term promoting sparsity and imposing structure at the same time.

The starting point is the *Elastic-Net*, a regression problem with

$$J(\mathbf{w}) = \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2$$

where the l_2 term is substituted by a new term $\lambda_G \|\mathbf{w}\|_G^2$. The new term can incorporate spatial and temporal information, e.g. using the discrete Laplacian.

Derivatives of the coefficients encourage smooth solutions (i.e., penalize roughness) while the l_1 term promotes sparse solutions.

---

[6]Grosenick, L. et al. *Neuroimage* **2013**, *72*, 304–321.

Structured Sparsity: TV-l_1[7]

The idea behind *Sparse Total Variation* (TV-l_1) is similar to Graph-Net.

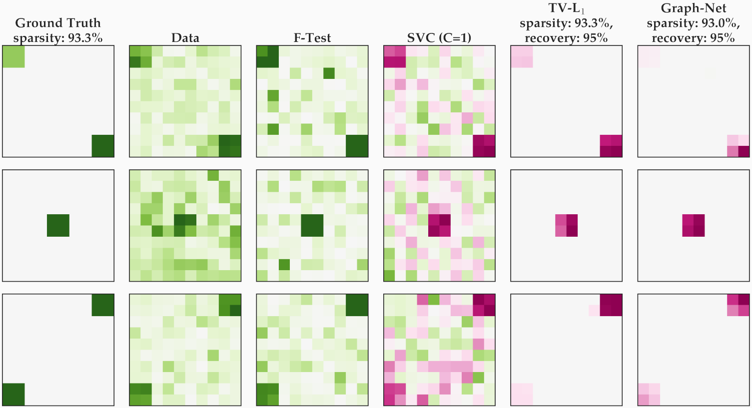
$$J(\mathbf{w}) = \lambda \left(\|\mathbf{w}\|_1 + \|\nabla \mathbf{w}\|_1 \right)$$

This time, the TV term, $\|\nabla \mathbf{w}\|_1$ favors sharp contours and piece-wise constant solutions to the regression problem, in contrast with the Graph-Net that prefers smoother solutions.

[7] Gramfort, A. et al. *2013 Int. Work. Pattern Recognit. Neuroimaging*.

Example: Noisy Data

A set of simulated noisy data consisting of 30 samples and 2 clases.



Example: Noisier Data

Same ground truth but noisier data.

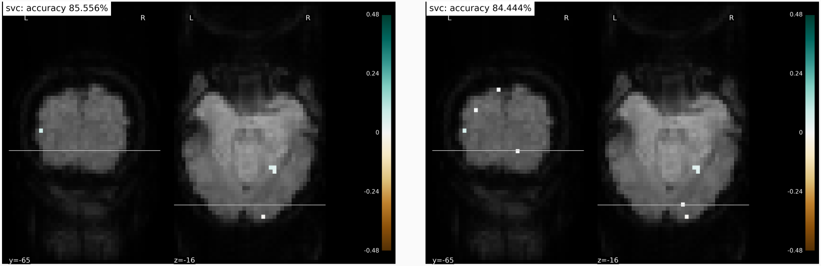
Regularization: Purpose

Regularization

- helps with overfitting,
- can do feature selection ($\rightarrow$ sparsity),
- and is necessary to solve the mathematical problem when the number of dimensions is very high (because it is ill-posed).

Example: Instability in Sparse Models

An example of a highly sparse model: SVC with l_1 regularization and Haxby's faces vs. houses



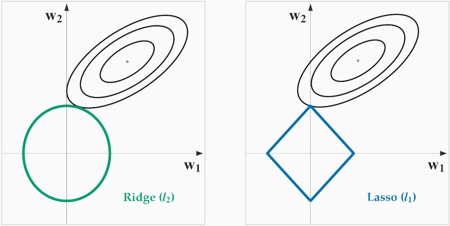
The outcome after running the same analysis with the same conditions: two different weight maps.

Regularization: How does it do it?

Regularization is a term, $J(\mathbf{w})$, added to the optimization problem:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} ||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \lambda J(\mathbf{w})$$

	$J(\mathbf{w})$	Effect
Ridge, l_2	$\|\|\mathbf{w}\|\|_2^2 = \sum_{i=1}^N w_i^2$	Shrinkage
Lasso, l_1	$\|\|\mathbf{w}\|\|_1 = \sum_{i=1}^N \|w_i\|$	Sparsity



Regularization: Effect on Coefficients