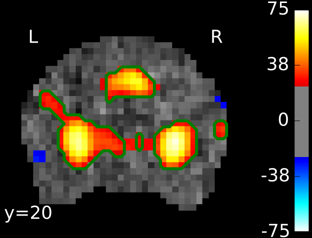
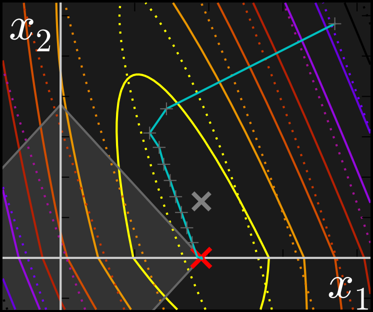
MVPA with SpaceNet: sparse and structured priors

Elvis DOHMATOB

Parietal Team, INRIA, Paris – France





1 Introducing the model

1 Brain decoding

- **We are given**:
  - $n = \#$ scans; $p =$ number of voxels in mask
  - design matrix: $X \in \mathbb{R}^{n \times p}$ (brain images)
  - response vector: $y \in \mathbb{R}^{n}$ (external covariates)
- Need to predict y on new data.
- Linear model assumption: $\mathbf{y} \approx \mathbf{X}\,\mathbf{w}$
- We seek to **estimate the weights map, w** that ensures best prediction / classification scores

1 The need for regularization

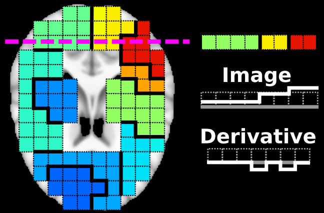
- **ill-posed problem**: high-dimensional ($n \ll p$)
- Typically $n \sim 10 - 10^3$ and $p \sim 10^4 - 10^6$
- We need **regularization** to reduce dimensions and encode practioner's priors on the weights $\mathbf{w}$

1 Why spatial priors ?

- **3D spatial gradient** (a linear operator)

$$\nabla : \mathbf{w} \in \mathbb{R}^p \longrightarrow (\nabla_x \mathbf{w}, \nabla_y \mathbf{w}, \nabla_z \mathbf{w}) \in \mathbb{R}^{p \times 3}$$

- penalize image grad ∇w
  $\Rightarrow$ regions
- Such priors are reasonable since **brain activity is spatially correlated**
- more stable maps and more predictive than unstructured priors (e.g SVM)
  [Hebiri 2011, Michel 2011, Baldassare 2012, Grosenick 2013, Gramfort 2013]

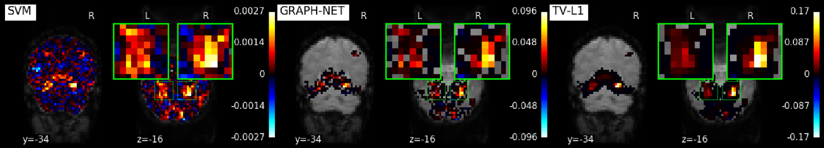


1 SpaceNet

- SpaceNet is a family of "**structure + sparsity**" priors for regularizing the models for brain decoding.
- SpaceNet generalizes
  - TV [Michel 2011],
  - Smooth-Lasso / GraphNet [Hebiri 2011, Grosenick 2013], and
  - TV-L1 [Baldassare 2012, Gramfort 2013].

- SpaceNet coefficients are more sparse and structured than SVM
-

1 In a nutshell



- SpaceNet coefficients are more sparse and structured than SVM
-

2 Methods

$$\mathbf{y} = \mathbf{X}\,\mathbf{w} + \text{“error”}$$

- Optimization problem (regularized model):

$$\textbf{minimize } \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \; + \; \text{penalty}(\mathbf{w})$$

- $\frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2$ is the **loss** term, and will be different for squared-loss, logistic loss, ...

2 The SpaceNet regularized model

- penalty($\mathbf{w}$) $= \alpha\Omega_\rho(\mathbf{w})$, where

$$\Omega_\rho(\mathbf{w}) := \rho\|\mathbf{w}\|_1 + (1-\rho)\begin{cases}\frac{1}{2}\|\nabla w\|^2, & \text{for GraphNet}\\ \|\mathbf{w}\|_{TV}, & \text{for TV-L1}\\ \dots \end{cases}$$

- α $(0 < \alpha < +\infty)$ is total amount regularization
- ρ $(0 < \rho \leq 1)$ is a mixing constant called the ℓ_1-**ratio**
  - $\rho = 1$ for Lasso

2 The SpaceNet regularized model

- $\text{penalty}(\mathbf{w}) = \alpha\Omega_\rho(\mathbf{w})$, where

$$\Omega_\rho(\mathbf{w}) := \rho\|\mathbf{w}\|_1 + (1-\rho)\begin{cases}\frac{1}{2}\|\nabla w\|^2, & \text{for GraphNet}\\ \|\mathbf{w}\|_{TV}, & \text{for TV-L1}\\ \ldots\end{cases}$$

- α $(0 < \alpha < +\infty)$ is total amount regularization
- ρ $(0 < \rho \leq 1)$ is a mixing constant called the ℓ_1**-ratio**
  - $\rho = 1$ for Lasso
- Problem is **convex**, **non-smooth**, and **heavily-ill-conditioned**.

- **Settings**: $\min f + g$; f smooth, g non-smooth
  f and g convex, ∇f L-Lipschitz; both f and g convex

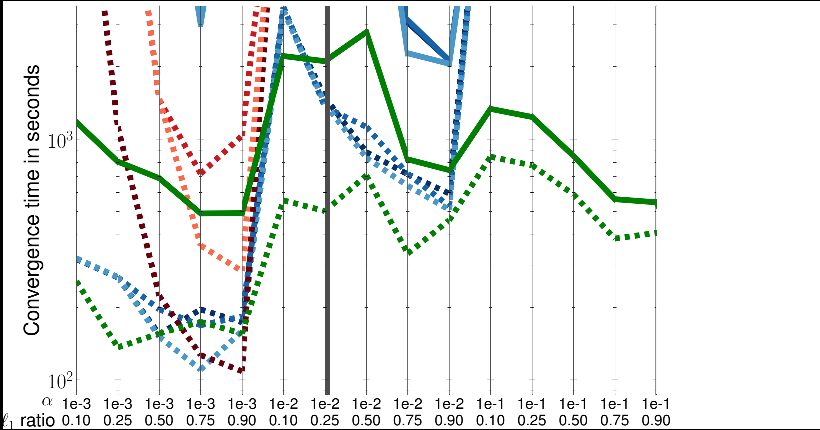
ISTA: $\mathcal{O}(\mathcal{L}_{\nabla f}/\epsilon)$ [Daubechies 2004]

Step 1: Gradient descent on f

Step 2: Proximal operator of g

FISTA: $\mathcal{O}(\mathcal{L}_{\nabla f}/\sqrt{\epsilon})$ [Beck Teboulle 2009]
$=$ ISTA with a "**Nesterov acceleration**" trick!

2 FISTA: Implementation for TV-L1



[DOHMATOB 2014 (PRNI)]

2 FISTA: Implementation for GraphNet

- Augment $\mathbf{X}$: $\tilde{X} := [X \quad c_{\alpha,\rho}\nabla]^T \in \mathbb{R}^{(n+3p)\times p}$
  $\Rightarrow \tilde{\mathbf{X}}\mathbf{z}^{(t)} = \mathbf{X}\mathbf{z}^{(t)} + c_{\alpha,\rho}\nabla(\mathbf{z}^{(t)})$

1. **Gradient descent step** (datafit term):
   $\mathbf{w}^{(t+1)} \leftarrow \mathbf{z}^{(t)} - \gamma\tilde{\mathbf{X}}^T(\tilde{\mathbf{X}}\mathbf{z}^{(t)} - \mathbf{y})$
2. **Prox step** (penalty term):
   $\mathbf{w}^{(t+1)} \leftarrow soft_{\alpha\rho\gamma}(\mathbf{w}^{(t+1)})$
3. **Nesterov acceleration**:
   $\mathbf{z}^{(t+1)} \leftarrow (1 + \theta^{(t)})\mathbf{w}^{(t+1)} - \theta^{(t)}\mathbf{w}^{(t)}$

Bottleneck: $\sim 80\%$ **of runtime** spent doing $\mathbf{X}z^{(t)}$!

- We badly need speedup!

2 Automatic model selection via Cross-Validation

- **Regularization parameters**:
$0 < \alpha_L < ... < \alpha_3 < \alpha_2 < \alpha_1 = \alpha_{max}$

- **Mixing constants**: $0 < \rho_M < ... < \rho_2 < \rho_1 \leq 1$

- Thus $L \times M$ grid to search over for best parameters

(α_1, ρ_1)	(α_1, ρ_2)	(α_1, ρ_3)	...	(α_1, ρ_M)
(α_2, ρ_1)	(α_2, ρ_2)	(α_2, ρ_3)	...	(α_2, ρ_M)
...	...	...	...	...
(α_L, ρ_1)	(α_L, ρ_2)	(α_L, ρ_3)	...	(α_L, ρ_M)

- CV Walks grid from **left to right** and **top to bottom** with **warm-staring**.

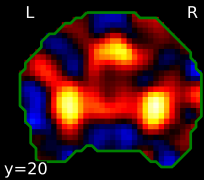
2 Automatic model selection via cross-validation

- The final model uses average of the the per-fold best weights maps (bagging)
- This bagging strategy ensures more stable and robust weights maps

2 Speedup via univariate screening

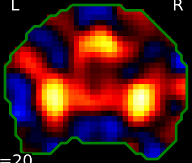
- Whereby we **detect and remove irrelevant voxels** before optimization problem is even entered!

2 $X^T y$ maps: relevant voxels stick-out

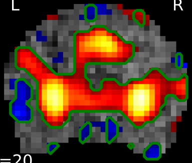


100% brain vol

2 $X^T y$ maps: relevant voxels stick-out

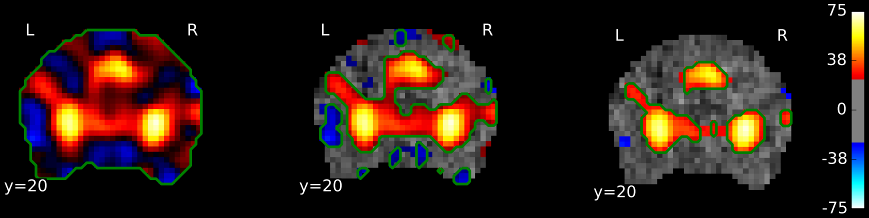


100% brain vol



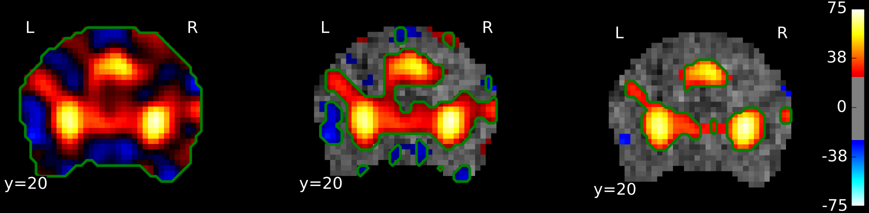
50% brain vol

2 $X^T y$ maps: relevant voxels stick-out



100% brain vol   50% brain vol   20% brain vol

2 $X^T y$ maps: relevant voxels stick-out

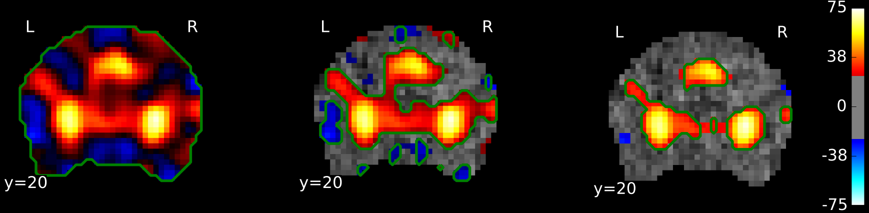


100% brain vol 50% brain vol 20% brain vol

- The 20% mask has the 3 bright blobs we would expect to get
- ... but contains much less voxels $\Rightarrow$ less run-time

2 Our screening heuristic

- $t_p :=$ pth percentile of the vector $|X^T y|$.
- Discard jth voxel if $|X_j^T y| < t_p$



$k = 100\%$ voxels $k = 50\%$ voxels $k = 20\%$ voxels

- Marginal screening [Lee 2014], but without the (invertibility) restriction $k \leq min(n, p)$.
- The regularization will do the rest...

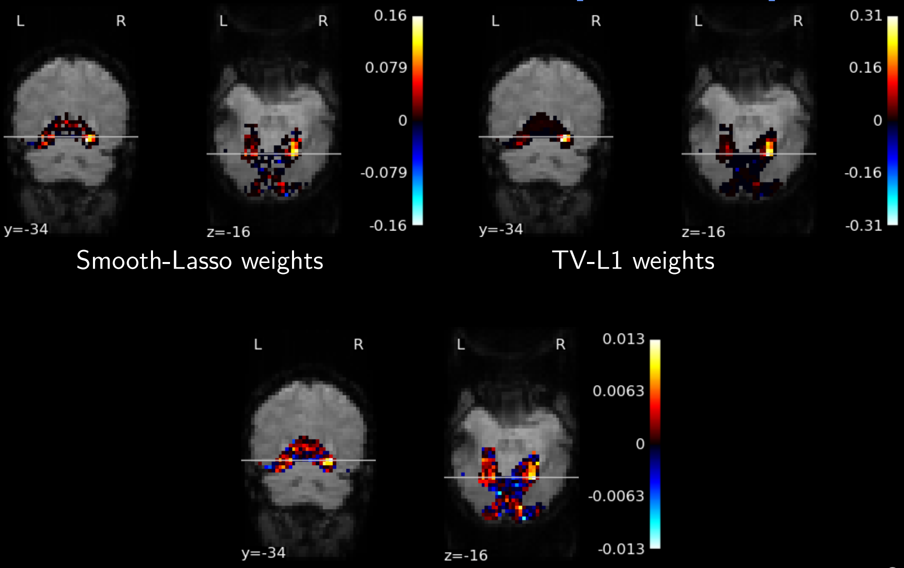
2 Our screening heuristic

- Our speedup heuristics produce upto 10-**fold speedup!**
- See [DOHMATOB 2015 (PRNI)] for a more detailed exposition of speedup heuristics developed.

3 Some experimental results

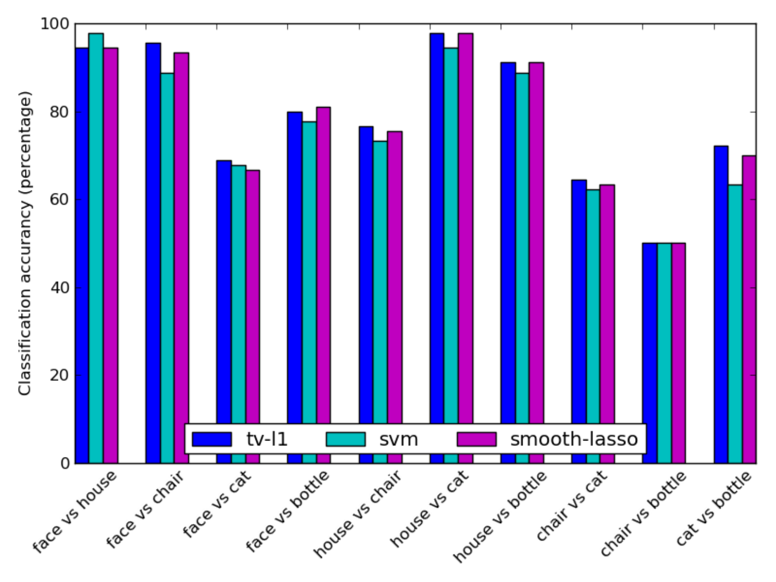
3 Weights: SpaceNet versus SVM

- Faces vs objects classification on [Haxby 2001]



SVM weights

3 Classification scores: SpaceNet versus SVM



3 Concluding remarks

- SpaceNet enforces both sparsity and structure, leading to better prediction / classification scores and more interpretable brain maps.

- The code runs (**on a laptop with 1 processor**) in $\sim$ 15 minutes for "simple" datasets, and $\sim$ 30 minutes for very difficult datasets.

- SpaceNet enforces both sparsity and structure, leading to better prediction / classification scores and more interpretable brain maps.

- The code runs (**on a laptop with 1 processor**) in $\sim$ 15 minutes for "simple" datasets, and $\sim$ 30 minutes for very difficult datasets.

- In the next release, SpaceNet will feature as part of Nilearn [Abraham et al. 2014] `http://nilearn.github.io`.

3 Interested in my work ?

Checkout:

- My home page at Parietal Team, INRIA: https://team.inria.fr/parietal/elvis/
- My Github page: https://github.com/dohmatob

3 Why $X^T y$ maps give a good relevance measure ?

- In an orthogonal design, least-squares solution is $\hat{\mathbf{w}}_{LS} = (X^T X)^{-1} X^T y = (I)^{-1} X^T y = X^T y$
  $\Rightarrow$ (intuition) $X^T y$ bears some info on optimal solution even for general $\mathbf{X}$

3 Why $X^T y$ maps give a good relevance measure ?

- In an orthogonal design, least-squares solution is $\hat{\mathbf{w}}_{LS} = (X^T X)^{-1} X^T y = (I)^{-1} X^T y = X^T y$
$\Rightarrow$ (intuition) $X^T y$ bears some info on optimal solution even for general $\mathbf{X}$
- Marginal screening: Set $S =$ indices of **top** k **voxels** j in terms of $|\mathbf{X}_j^T \mathbf{y}|$ values
  - In [Lee 2014], $k \leq min(n, p)$, so that $\hat{\mathbf{w}}_{LS} \sim (\mathbf{X}_S^T \mathbf{X}_S)^{-1} \mathbf{X}_S^T \mathbf{y}$
  - We don't require invertibility condition $k \leq min(n, p)$. Our spatial regularization will do the rest!

3 Why $X^T y$ maps give a good relevance measure ?

- In an orthogonal design, least-squares solution is $\hat{\mathbf{w}}_{LS} = (X^T X)^{-1} X^T y = (I)^{-1} X^T y = X^T y$
  $\Rightarrow$ (intuition) $X^T y$ bears some info on optimal solution even for general $\mathbf{X}$
- Marginal screening: Set S = indices of **top** k **voxels** j in terms of $|\mathbf{X}_j^T \mathbf{y}|$ values
  - In [Lee 2014], $k \leq min(n, p)$, so that $\hat{\mathbf{w}}_{LS} \sim (\mathbf{X}_S^T \mathbf{X}_S)^{-1} \mathbf{X}_S^T \mathbf{y}$
  - We don't require invertibility condition $k \leq min(n, p)$. Our spatial regularization will do the rest!
- Lots of **screening rules** out there: [El Ghaoui 2010, Liu 2014, Wang 2015, Tibshirani 2010, Fercoq 2015]