Social-sparsity brain decoders: faster spatial sparsity

Gaël VAROQUAUX*, Matthieu KOWALSKI*†, Bertrand THIRION*,
*INRIA Parietal, Neurospin, bât 145, CEA Saclay, 91191 Gif sur Yvette, France
firstname.lastname@inria.fr
†L2S, Univ Paris-Sud – CNRS – CentraleSupelec

***Abstract*—Spatially-sparse predictors are good models for brain decoding: they give accurate predictions and their weight maps are interpretable as they focus on a small number of regions. However, the state of the art, based on total variation or graph-net, is computationally costly. Here we introduce sparsity in the local neighborhood of each voxel with social-sparsity, a structured shrinkage operator. We find that, on brain imaging classification problems, social-sparsity performs almost as well as total-variation models and better than graph-net, for a fraction of the computational cost. It also very clearly outlines predictive regions. We give details of the model and the algorithm.**

***Index Terms*—brain decoding, sparsity, spatial regularization**

I. Introduction: Spatial sparsity in decoding

Machine learning can predict behavior or phenotype from brain images. Across subjects, it can give indications on a pathology or its progression, for instance capturing atrophy patterns of Alzheimer-related cognitive decline from anatomical imaging [1]. To study brain function, it has been used extensively to predict some cognitive parameters associated with the presented stimuli from functional brain images such as functional Magnetic Resonance Imaging (fMRI) [2]. In these applications, the number of samples is small (hundreds or less), while the number of features is typical the number of voxels in the brain, 50 000 or more. The estimation is ill-posed, as there are much more parameters to estimate than the number of samples, which calls for regularization.

In brain decoding, good prediction is important, but also retrieving and understanding the aspects of brain images that drive this prediction. Linear models, as linear support vector machines (SVM), are often used because they work well in the low-sample regime. An additional benefit of these models is that their weights form brain maps [3]. However, suitable regularization is necessary for these weights to retrieve well the important regions [4], [5]. Sparse penalties select voxels [6], [7], but only a subset of the important ones [4]. Spatial and sparse penalties, using total variation (TV) [8] or graph-net [9], help the decoder capture full regions [5]. TV and its variants are state-of-the-art regularizers for brain images. In addition to decoding [8], [10], [5], they have been used with great success for applications as diverse as MR image reconstruction [11], super-resolution [12], prediction from anatomical images [13], and even regularization of spatial registration [14]. The main drawback of spatial sparsity as in TV and related penalties is its computational cost.

Here we introduce spatial sparsity based on "social sparsity", a relaxed structured penalty with a simple closed-form shrinkage operator. We first detail the mathematical underpinnings of the model and then perform decoding experiments that show that the model is almost as accurate in prediction as the TV-based state of the art and much faster.

Notations: Vectors are written in bold: $\mathbf{x}$. Indexing vectors is written as sub-scripts: $\mathbf{x}_i$. In an iterative scheme, the iteration number is a sub-script in parenthesis: $\mathbf{x}_{(k)}$.

II. Methods: Introducing social sparsity

A. Spatial penalties for neuroimaging

A sparsity assumption on the decoders maps has been identified early as a means to select voxels relevant for decoding [6], [7]. Sparsity has been used with great success in statistics, signal processing, and machine learning. Indeed, it combines good properties for prediction and denoising when the ground truth is actually sparse [15]. Sparse models can be estimated by adding to the data-fit term an ℓ_1 penalty for sparsity, as in the famed Lasso. The Iterative Shrinkage/Thresholding Algorithm (ISTA) is a common algorithm to solve the corresponding minimization problem [16]. It alternates a gradient-descent step for the data-fit energy and, for sparsity, a *soft-thresholding*: a proximal operator which corresponds to the Euclidean projection on some ℓ_1 ball:

$$\ell_1 \text{ prox of } \mathbf{w}, \text{ scale } \lambda: \quad \forall i \quad \mathbf{w}_i \leftarrow \mathbf{w}_i \left(1 - \frac{\lambda}{|\mathbf{w}_i|}\right)^+ \quad (1)$$

where the operation is applied element-wise –for every coordinate $\mathbf{w}_i$ of $\mathbf{w}$– and $(.)^+$ is the positive part. This operation is reminiscent of element-wise thresholding: it sets to zero all the entries of $\mathbf{w}$ that are smaller than λ in absolute value, and shrinks by λ the remaining.

However, on brain images, simple sparse models as with the ℓ_1 penalty select a subset of the important voxels [4], [17], [18]. Indeed as the information in a voxel is very correlated to its neighbors, sparsity focuses on one representative in a local neighborhood [4]. Improved penalties add a spatial term to couple neighbors and create structured sparsity. Total-variation (TV) imposes sparsity in the image gradients [8]. Coupled with sparsity in a TV-ℓ_1 penalty it recovers very well the predictive brain regions [10], [5]. It is the state of the art for brain decoding in terms of predictive power and of interpretability of the decoder maps. The main drawback of total variation is that it leads to very slow solvers. Indeed, it cannot be formulated in terms of a thresholding-like operator. Its proximal operator is computational costly as it couples all

the voxels. Another spatial penalty, related but faster, Graph-net, imposes smoothness, rather than sparsity, on the image gradients [9]. Because it is a differentiable penalty, it leads to faster optimization, though it imposes less spatial structure.

The challenges of solving TV and related penalties arise from their very strength: they impose sparsity not on the voxels of the image, but on a representation capturing differences between voxels. This concept is an instance of *analysis sparsity*, which leads to other penalties successful for predicting from brain images [19]. Overlapping group sparsity is another analysis penalty which has been heavily used in image processing [20]: rather than putting voxels to zeros, it penalizes a full local neighborhood, using a penalty on blocks. As a voxel is in several neighborhoods, these blocks are overlapping. This overlap leads to an optimization problem that has the same structure –and same cost– as that of TV-ℓ_1.

All these estimators, and related optimization problems, can be solved with an ISTA algorithm, or its accelerated variant the FISTA. In fact, these approaches are the best option in the case of brain decoding [21] although the TV-ℓ_1 proximal operator involves a costly sub-iteration. In the case of non-overlapping blocks, group sparsity is simply ℓ_{21} penalty applied on each group, as in the famed group-lasso. The proximal operator is closed-form and performs a soft-thresholding on each group based on its Euclidean norm:

$$\underset{\text{on group } \mathcal{G}}{\ell_{21} \text{ prox}} : \quad \forall i \in \mathcal{G} \quad \mathbf{w}_i \leftarrow \mathbf{w}_i \left(1 - \frac{\lambda}{\sqrt{\sum_{j\in\mathcal{G}} \mathbf{w}_j^2}}\right)^+ \quad (2)$$

for a coordinate $\mathbf{w}_i$ in the group $\mathcal{G}$. As with soft-thresholding for the ℓ_1 penalty, the operation is applied element-wise for all the groups and leads to fast optimizations.

B. Fast spatial structure with social sparsity

There is a large gap in computational cost between sparsity imposed on separate items, coordinates or groups, and coupling the sparsity of a voxel to that of its neighbor, as in overlapping group sparsity or penalties related to TV. We introduce social sparsity [22] which is a tradeoff between the two scenarios. It applies a soft-thresholding similar to group-lasso, using the norm of a local neighborhood, but modifies only the coordinate $\mathbf{w}_i$ at the center of this neighborhood. In a sense, social sparsity "forgets" overlaps across neighborhoods. This makes solvers much faster, as we show in the following.

a) The social-sparsity operator: We focus here on the most popular "social-sparse" operator: the windowed group-lasso. For each element $\mathbf{w}_i$, we associate a neighborhood $\mathcal{N}(i)$ of coordinates. The shrinkage operator reads

$$\underset{\text{shrinkage}}{S(\mathbf{w},\lambda)\ \text{social}} : \quad \forall i \quad \mathbf{w}_i \leftarrow \mathbf{w}_i \left(1 - \frac{\lambda}{\sqrt{\sum_{j\in\mathcal{N}(i)} \alpha_j^i \mathbf{w}_j^2}}\right)^+ \quad (3)$$

where the α_j^i are weights representing the shape of the neighborhood (the simplest choice being the rectangular window: all α_j^i are equal).

An interesting interpretation of this operator, is that it can be seen as a fast approximation of the group-lasso with overlaps as presented in [23]. Indeed, it can be shown that the social-sparse operator is equivalent to applying a regular group-lasso operator in a high dimensional space where all the variables are duplicated in order to form independent groups (there are as many groups as variables). Then the result is projected back in the original space following an oblique projection [22]. In practice, similar performances are obtained without the cost encountered by the group-lasso with overlaps.

Algorithm 1: FISTA: social sparsity to estimate $\mathbf{w}$

input : Initialization $\mathbf{w}_{(0)} \in \mathbb{R}^p$, penalization amount λ, L-Lipschitz gradient of loss ∇F

$\mathbf{v}_{(1)} \leftarrow \mathbf{w}_{(0)}, \qquad k \leftarrow 1, \qquad t_{(0)} \leftarrow 0;$

while $\|\mathbf{w}_{(k)} - \mathbf{w}_{(k-1)}\|_\infty > tol\|\mathbf{w}_{(k)}\|_\infty$ **do**

$k \leftarrow k+1, \qquad t_k \leftarrow \frac{1}{2}\big(1+\sqrt{1+4t_{(k-1)}^2}\big);$

$\mathbf{w}_{(k)} \leftarrow S\big(\mathbf{v}_{(k)} - \frac{1}{L}\nabla F(\mathbf{v}_{(k)}),\ \frac{\lambda}{L}\big) \quad S$ given by (3);

$\mathbf{v}_{(k)} \leftarrow \mathbf{w}_{(k)} + \frac{t_{(k-1)}-1}{t_{(k)}}(\mathbf{w}_{(k)} - \mathbf{w}_{(k-1)});$

b) Choice of the neighborhood: Coupling neighboring voxels is crucial, as demonstrated by the success of spatial penalties in brain imaging. However the spatial extent of that coupling should be small in brain imaging. Indeed, the typical scale used to smooth fMRI data is of 6 mm, for 3 mm voxels. Similarly, anatomical images, with voxels of 1 mm, are often smoothed with a kernel of 2 mm. We use as a neighborhood $\mathcal{N}(i)$ of a voxel i its 6 immediate neighbors. To let the behavior of a group be driven more by its central voxel, we set the relative weights α of the 6 other voxels to .7.

c) A FISTA solver: The social-sparsity shrinkage S is not the proximal operator of a known penalty. Yet, it has been shown to yield good estimations in proximal optimization schemes [22]. We use a FISTA, an accelerated variant of ISTA, as for TV-ℓ_1 [21], [19]. The brain-imaging data fit appears via a loss $L : \mathbf{w} \in \mathbb{R}^p \rightarrow \mathbb{R}$, the gradient of which should be Lipschitz-continuous. Typically, the logistic loss is used for classification problems and the squared loss for regression. For completeness we detail the scheme in Algorithm 1.

d) Parameter selection: We set the regularization parameter λ by nested cross-validation, using the same strategies as can be used in Graph-net or TV-ℓ_1 to speed up computation [24]. We do 8 folds. For each fold, we scan the λ parameter from large values to small values, with warm start of the solver. As the model is similar to an ℓ_1 penalty but with additional constraints, values of λ that lead pure ℓ_1 models to be fully sparse will also give fully sparse social-sparsity models. Hence we start our path at λ_{max}, the largest λ giving non-empty ℓ_1 model[1]. We visit 5 values on a logarithmic scale from λ_{max} to $\frac{1}{20}\lambda_{\text{max}}$. As in Graph-net or TV-$\ell_1$ solvers [24], we do early stopping of the optimizer on the left-out prediction error during

[1] $\lambda_{\text{max}} = \|\mathbf{X}^\mathsf{T}\mathbf{y}\|_\infty$ for lasso, and $\lambda_{\text{max}} = \frac{1}{n}\|\mathbf{X}^\mathsf{T}\tilde{\mathbf{y}}\|_\infty$ for ℓ_1 logistic, where $\tilde{\mathbf{y}}$ is the weighted output vector: $\tilde{\mathbf{y}}_i = \frac{n^+}{n}$ for samples in the positive class, and $\tilde{\mathbf{y}}_i = \frac{n^-}{n}$ for samples in the negative class, with n the number of samples, n^+ (resp. n^-) the number in the positive (resp. negative) class.

parameter selection. The final coefficients are the average of the coefficient for the optimal λ on each of the 8 folds.

Finally, as it can be done with graph-net and TV-ℓ_1, we use univariate feature screening, retaining 20% of the features. The motivation for this screening is that sparse models are highly likely to put the corresponding features to zero [24].

III. Experiments: Accuracy and run time

Datasets: We study social sparsity on publicly-available datasets for two applications: intra-subject brain-decoding from fMRI, and inter-subject prediction from voxel-based morphometry (VBM).

For fMRI brain decoding, we use a standard visual-object recognition dataset [2]. We perform on all 5 subjects intra-subject 2-class decoding of 14 pairs of stimuli of varying difficulties (listed on Figure 1). To measure prediction accuracy, we perform cross-validation, leaving out 2 of the 12 sessions.

For VBM inter-subject prediction, we use the OASIS anatomical imaging dataset [25]. We predict gender and use a cross-validation of random splits of 20% of the subjects.

Experimental settings: On all classification tasks, we fit a model with a logistic loss and measure prediction accuracy as well as wall time in ten iterations of cross-validation.

We compare social sparsity to graph-net and TV-ℓ_1, as implemented in the Nilearn library, as well as the most commonly-used decoder, a linear SVM with 20% univariate feature selection. To fully replicate common practice we use the default $C = 1$ for the SVM, rather than cross-validation. All models, including social-sparsity, are implement in Python, using scikit-learn for the SVM [26]. The graph-net and TV-ℓ_1 implementation use the same feature-selection and path strategies as our social-sparsity solver to speed up computation. An important detail for computation time is the stopping criteria of the algorithms. We use the same for Graph-net, TV-ℓ_1, and social sparsity: a 10^{-4} cutoff on the relative maximum change on the weights $\mathbf{w}$ (see Alg. 1).

IV. Results: Striking a good tradeoff

Experiments outline a tradeoff between prediction accuracy and computation time. Fig. 1 displays the relative prediction accuracy and run time. TV-ℓ_1 predicts best on average over the various classification tasks. However, it is followed closely by social sparsity which outperforms graph-net[2]. The SVM performs much worse than the spatial sparsity, aside from the VBM data where we find that all models perform similarly.

In terms of run time, we find that graph-net is on average 4 times faster than TV-ℓ_1, but social sparsity is 3 times faster than graph-net. The SVM-based decoder is 20 times faster than social sparsity, *ie* 240 times faster than TV-ℓ_1.

Finally, an important aspect of the brain decoders is whether they segment well the brain regions that support the decoding. Such a question is hard to validate, yet there is evidence that TV-ℓ_1 is a good approach [5]. Fig. 2 displays the decoder maps for the object-recognition tasks. For these tasks, we expect

[2]All differences are significant in a Wilcoxon rank test.

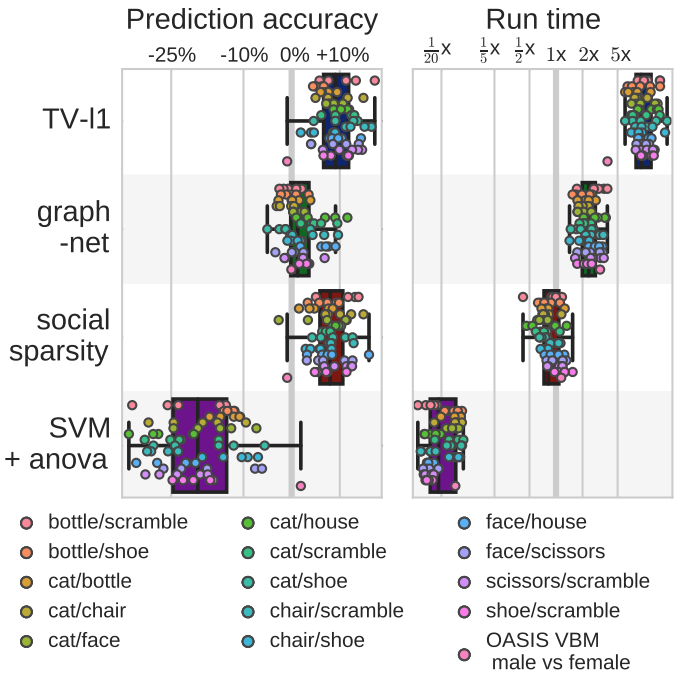

Fig. 1. **Prediction accuracy and computation time** for different classification tasks from brain images: 14 on the Haxby dataset, and gender prediction on OASIS. Values are displayed relative to the mean over 4 classifiers: TV-ℓ_1, graph-net, social sparsity, and a linear SVM with 20% univariate feature selection. Each subject of the Haxby study gives one data point.

prediction to be driven by the functional areas of the visual cortex [27]. Indeed, the maps outline regions in known visual areas. The graph-net maps are much more scattered and less structured than the others. Conversely, the social sparsity maps are sparser and outline a smaller number of clusters.

V. Conclusion: Be social

Brain decoders benefit strongly from spatial sparsity that helps them narrow on regions important for prediction. Total variation is a powerful and principled solution, but it comes with a hefty computational cost. Social sparsity can be used to introduce penalties on local neighborhoods in the image. It is more heuristic, as it does not minimize a known convex cost. We find empirically that it performs very slightly less well that TV-ℓ_1 in terms of prediction accuracy, but is more than ten times faster. The corresponding decoder maps are more sparse and focus on a smaller number of regions than TV-ℓ_1. Social sparsity outperforms graph-net, the faster contender to TV-ℓ_1, on speed, accuracy, and interpretability. We have found that it strikes a very interesting balance for brain decoding: much faster and almost as good for prediction as TV-ℓ_1. A full social-sparsity model-fit with hyper-parameter selection takes only 20 times longer than a simple SVM with default hyper-parameters. With social-sparsity, spatially-structured brain decoders are fast: typically 30 s with parameter selection on a 2 GHz i7 CPU.

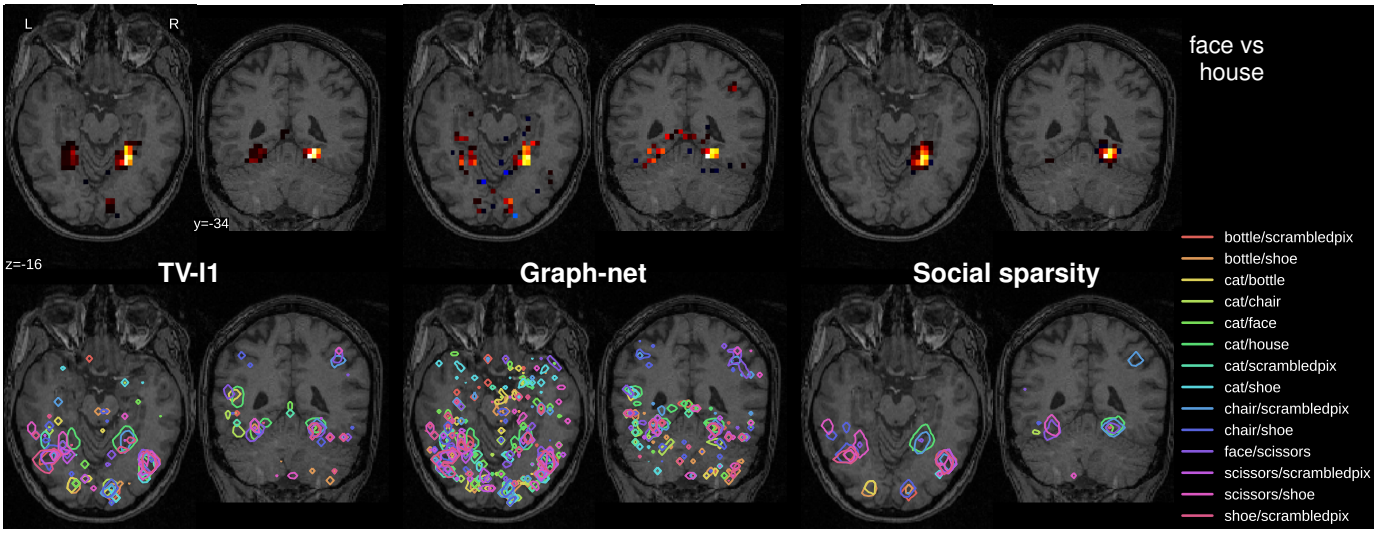

Fig. 2. **Decoder maps for the object-classification task – Top**: weight maps for the face-versus-house task. Overall, the maps segment the right and left parahippocampal place area (PPA), a well-known place-specific regions, although the left PPA is weak in TV-ℓ_1, spotty in graph-net, and absent in social sparsity. **Bottom**: outlines at 0.01 of the other tasks. Beyond the PPA, several known functional regions stand out such as primary or secondary visual areas around the prestriate cortex as well as regions in the lateral occipital cortex, responding to structured objects [27]. Note that the graph-net outlines display scattered small regions even thought the value of the contours is chosen at 0.01, well above numerical noise.

Acknowledgment: OASIS was supported by grants P50 AG05681, P01 AG03991, R01 AG021910, P50 MH071616, U24 RR021382, R01 MH56584. The authors acknowledge funding from the EU FP7/2007-2013 under grant agreement 604102 (HBP).

References

[1] Y. Fan, N. Batmanghelich, C. M. Clark, C. Davatzikos, and ADNI, "Spatial patterns of brain atrophy in mci patients, identified via high-dimensional pattern classification, predict subsequent cognitive decline," *Neuroimage*, vol. 39, p. 1731, 2008.

[2] J. Haxby, I. Gobbini, M. Furey *et al.*, "Distributed and overlapping representations of faces and objects in ventral temporal cortex," *Science*, vol. 293, p. 2425, 2001.

[3] J. Mourão-Miranda, A. L. Bokde, C. Born, H. Hampel, and M. Stetter, "Classifying brain states and determining the discriminating activation patterns: Support vector machine on functional MRI data," *NeuroImage*, vol. 28, p. 980, 2005.

[4] G. Varoquaux, A. Gramfort, and B. Thirion, "Small-sample brain mapping: sparse recovery on spatially correlated designs with randomization and clustering," *ICML*, 2006.

[5] A. Gramfort, B. Thirion, and G. Varoquaux, "Identifying predictive regions from fMRI with TV-L1 prior," in *PRNI*, 2013, p. 17.

[6] O. Yamashita, M. aki Sato, T. Yoshioka, F. Tong, and Y. Kamitani, "Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns," *NeuroImage*, vol. 42, p. 1414, 2008.

[7] M. K. Carroll, G. A. Cecchi, I. Rish, R. Garg, and A. R. Rao, "Prediction and interpretation of distributed neural activity with sparse models," *NeuroImage*, vol. 44, p. 112, 2009.

[8] V. Michel, A. Gramfort, G. Varoquaux *et al.*, "Total variation regularization for fMRI-based prediction of behavior," *IEEE Trans Med Im*, vol. 30, p. 1328, 2011.

[9] L. Grosenick, B. Klingenberg, K. Katovich *et al.*, "Interpretable whole-brain prediction analysis with graphnet," *NeuroImage*, vol. 72, p. 304, 2013.

[10] L. Baldassarre, J. Mourao-Miranda, and M. Pontil, "Structured sparsity models for brain decoding from fMRI data," in *PRNI*, 2012, p. 5.

[11] J. Huang, S. Zhang, and D. Metaxas, "Efficient MR image reconstruction for compressed MR imaging," *Med. Image Anal.*, vol. 15, p. 670, 2011.

[12] S. Tourbier, X. Bresson, P. Hagmann, J.-P. Thiran, R. Meuli, and M. B. Cuadra, "An efficient total variation algorithm for super-resolution in fetal brain MRI with adaptive regularization," *NeuroImage*, vol. 118, p. 584, 2015.

[13] M. Dubois, F. Hadj-Selem, T. Lofstedt *et al.*, "Predictive support recovery with TV-elastic net penalty and logistic regression: an application to structural MRI," in *PRNI*, 2014.

[14] J.-B. Fiot, H. Raguet, L. Risser, L. D. Cohen, J. Fripp, F.-X. Vialard, and ADNI, "Longitudinal deformation models, spatial regularizations and learning strategies to quantify alzheimer's disease progression," *NeuroImage: Clinical*, vol. 4, p. 718, 2014.

[15] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity: the lasso and generalizations*. CRC Press, 2015.

[16] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, p. 1413, 2004.

[17] I. Rish, G. Cecchi, K. Heuton, M. Baliki, and A. V. Apkarian, "Sparse regression analysis of task-relevant information distribution in the brain," in *SPIE Medical Imaging*, 2012, p. 831412.

[18] J. M. Rondina, T. Hahn, L. De Oliveira *et al.*, "Scors, a method based on stability for feature selection and mapping in neuroimaging," *Transactions on Medical Imaging*, vol. 33, p. 85, 2014.

[19] M. Eickenberg, E. Dohmatob, B. Thirion, and G. Varoquaux, "Total variation meets sparsity: statistical learning with segmenting penalties," *MICCAI*, 2015.

[20] J. Mairal, F. Bach, and J. Ponce, *Sparse Modeling for Image and Vision Processing*. Now Publishers, 2014.

[21] E. Dohmatob, A. Gramfort, B. Thirion, and G. Varoquaux, "Benchmarking solvers for TV-l1 least-squares and logistic regression in brain imaging," *PRNI*, 2014.

[22] M. Kowalski, K. Siedenburg, and M. Dorfler, "Social sparsity! neighborhood systems enrich structured shrinkage operators," *Transactions on Signal Processing*, vol. 61, p. 2498, 2013.

[23] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *ICML*, 2009, p. 433.

[24] E. Dohmatob, M. Eickenberg, B. Thirion, and G. Varoquaux, "Speeding-up model-selection in graphnet via early-stopping and univariate feature-screening," in *PRNI*, 2015, pp. 17–20.

[25] D. S. Marcus, T. H. Wang, J. Parker *et al.*, "Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults." *J Cogn Neurosci*, vol. 19, p. 1498, 2007.

[26] A. Abraham, F. Pedregosa, M. Eickenberg *et al.*, "Machine learning for neuroimaging with scikit-learn," *Frontiers in neuroinformatics*, vol. 8, 2014.

[27] K. Grill-Spector and R. Malach, "The human visual cortex," *Annu. Rev. Neurosci.*, vol. 27, p. 649, 2004.