

AUTHOR QUERY FORM

 ELSEVIER	Journal: YNIMG	Please e-mail or fax your responses and any corrections to: E-mail: Corrections.ESSD@elsevier.spitech.com Fax: +1 619 699 6721
	Article Number: 10064	

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult <http://www.elsevier.com/artworkinstructions>.

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

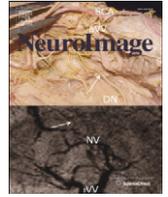
Location in article	Query / Remark: click on the Q link to go Please insert your reply or correction at the corresponding line in the proof
Q1	Please confirm that given names and surnames have been identified correctly.
Q2	Figure 1 contain(s) text below the minimum required font size of 6pts and there is no sufficient space available for the text to be enlarged. Please confirm if we can re-type for you, which might change the current location of data or image. Otherwise, please provide replacement figure file. For more information, visit Elsevier website at: http://www.elsevier.com/framework_authors/artwork/artwork.pdf .
Q3, Q4	Please provide a definition for the significance of bold entries in the table.
Q5, Q8, Q9, Q10, Q11, Q12, Q13	The citation “Karmarkar et al., 2012” has been changed to match the author name/date in the reference list. Please check here and in subsequent occurrences, and correct if necessary.
Q6	“Section 2.3” and other references to numbered sections were replaced because the section titles were modified from numbered to unnumbered as specified by the Journal.
Q7	“Section 2.1.5” was replaced because the section titles were modified from numbered to unnumbered as specified by the Journal.
Q14	Please check the page range in Ref. Donoho, 2006.
Q15	Please provide the volume number and page range for the bibliography in Ref. [Grosenick et al., 2010].
Q16	Please provide an update for reference “Karmarkar et al., submitted for publication”.

Please check this box if you have no corrections to make to the PDF file.

Thank you for your assistance.

Contents lists available at [SciVerse ScienceDirect](#)

NeuroImage

journal homepage: www.elsevier.com/locate/ynimg

Highlights

Interpretable whole-brain prediction analysis with GraphNet*NeuroImage xxx (2013) xxx–xxx*Logan Groseⁿick ^{a,b,*}, Brad Klingenberg ^b, Kiefer Katovich ^c, Brian Knutson ^c, Jonathan E. Taylor ^b^a Center for Mind, Brain, and Computation, Stanford University, Stanford, CA, USA^b Department of Statistics, Stanford University, Stanford, CA, USA^c Department of Psychology, Stanford University, Stanford, CA, USA

- ▶ We introduce robust, interpretable models for prediction with whole-brain fMRI data. ▶ These use a sparsity-inducing penalty that automatically selects important voxels. ▶ They also include a graph penalty to structure the solution. ▶ They outperform state-of-the-art classifiers on whole-brain fMRI data.
- ▶ They predict outcomes on new data collected years after the data used for training.



Interpretable whole-brain prediction analysis with GraphNet

Logan Grosenick^{a,b,*}, Brad Klingenberg^b, Kiefer Katovich^c, Brian Knutson^c, Jonathan E. Taylor^b

^a Center for Mind, Brain, and Computation, Stanford University, Stanford, CA, USA

^b Department of Statistics, Stanford University, Stanford, CA, USA

^c Department of Psychology, Stanford University, Stanford, CA, USA

ARTICLE INFO

Article history:

Accepted 26 December 2012

Available online xxxx

ABSTRACT

Multivariate machine learning methods are increasingly used to analyze neuroimaging data, often replacing more traditional “mass univariate” techniques that fit data one voxel at a time. In the functional magnetic resonance imaging (fMRI) literature, this has led to broad application of “off-the-shelf” classification and regression methods. These generic approaches allow investigators to use ready-made algorithms to accurately decode perceptual, cognitive, or behavioral states from distributed patterns of neural activity. However, when applied to correlated whole-brain fMRI data these methods suffer from coefficient instability, are sensitive to outliers, and yield dense solutions that are hard to interpret without arbitrary thresholding. Here, we develop variants of the Graph-constrained Elastic Net (GraphNet), a fast, whole-brain regression and classification method developed for spatially and temporally correlated data that automatically yields interpretable coefficient maps (Grosenick et al., 2009b). GraphNet methods yield sparse but structured solutions by combining structured graph constraints (based on knowledge about coefficient smoothness or connectivity) with a global sparsity-inducing prior that automatically selects important variables. Because GraphNet methods can efficiently fit regression or classification models to whole-brain, multiple time-point data sets and enhance classification accuracy relative to volume-of-interest (VOI) approaches, they eliminate the need for inherently biased VOI analyses and allow whole-brain fitting without the multiple comparison problems that plague mass univariate and roaming VOI (“searchlight”) methods. As fMRI data are unlikely to be normally distributed, we (1) extend GraphNet to include robust loss functions that confer insensitivity to outliers, (2) equip them with “adaptive” penalties that asymptotically guarantee correct variable selection, and (3) develop a novel sparse structured Support Vector GraphNet classifier (SVGN). When applied to previously published data (Knutson et al., 2007), these efficient whole-brain methods significantly improved classification accuracy over previously reported VOI-based analyses on the same data (Grosenick et al., 2008; Knutson et al., 2007) while discovering task-related regions not documented in the original VOI approach. Critically, GraphNet estimates fit to the Knutson et al. (2007) data generalize well to out-of-sample data collected more than three years later on the same task but with different subjects and stimuli (Karmarkar et al., submitted for publication). By enabling robust and efficient selection of important voxels from whole-brain data taken over multiple time points (> 100,000 “features”), these methods enable data-driven selection of brain areas that accurately predict single-trial behavior within and across individuals.

© 2012 Published by Elsevier Inc.

Introduction

Accurately predicting subject behavior from functional brain data is a central goal of neuroimaging research. In functional magnetic resonance imaging (fMRI) studies, investigators measure the blood oxygen-level dependent (BOLD) signal—a proxy for neural activity—and relate this signal to psychophysical or psychological variables of interest. Historically, modeling is performed one voxel at a time to yield a map of univariate statistics that are then thresholded according to some heuristic to yield a “brain map” suitable for visual

inspection. Over the past decade, however, a growing number of neuroimaging studies have applied machine learning analyses to fMRI data to model effects across multiple voxels. Commonly referred to as “multivariate pattern analysis” (Hanke et al., 2009) or “decoding” (to distinguish them from more commonly-used “mass-univariate” methods (Friston et al., 1995)), these approaches have allowed investigators to use activity patterns across multiple voxels to classify image categories during visual presentation (Peelen et al., 2009; Shinkareva et al., 2008), image categories during memory retrieval (Polyn et al., 2005), intentions to move (Haynes et al., 2007), and even intentions to purchase (Grosenick et al., 2008) (to name just a few applications—see also (Bray et al., 2009; Haynes and Rees, 2006; Norman et al., 2006; O’Toole et al., 2007; Pereira et al., 2009), and examples in *NeuroImage* Volume 56 Issue 2). In multiple cases,

* Corresponding author at: 220 Panama Street, Ventura Hall Rm. 30, Stanford, CA 94305. Fax: +1 650 283 0010.

E-mail address: logang@gmail.com (L. Grosenick).

these statistical learning algorithms have shown better predictive performance than standard mass-univariate analyses (Haynes and Rees, 2006; Pereira et al., 2009).

Despite these advances, analysis of neuroimaging data with statistical learning algorithms is still young. Most of the research that has applied statistical learning algorithms to fMRI data has been conducted by a few laboratories (Norman et al., 2006), and most analyses have been conducted with off-the-shelf classifiers (Norman et al., 2006; Pereira et al., 2009, but cf. Brodersen et al., 2011; Chappell et al., 2009; Grosenick et al., 2008; Hutchinson et al., 2009; Michel et al., 2011; Ng et al., 2012). These classifiers are often applied to volume of interest (VOI) data within subjects rather than whole-brain data across subjects (Etzel et al., 2009; Pereira et al., 2009, but cf. Grosenick et al., 2009b, 2010; Michel et al., 2011; Mitchell et al., 2004; Mourão-Miranda et al., 2007; Ng et al., 2012; Ryali et al., 2010; van Gerven and Heskes, 2012). While these classifiers have a venerable history in the machine learning literature, they were not originally developed for application to whole-brain neuroimaging data, and so suffer from inefficiencies in this context. Specifically, the large number of features (usually voxel data) and spatiotemporal correlations characteristic of fMRI data present unique challenges for off-the-shelf classifiers.

Indeed, the purpose of off-the-shelf classifiers in the machine learning literature (e.g., discriminant analysis (DA), naive Bayes (NB), k-nearest neighbors (kNN), random forests (RF), and support vector machines (SVM)) has been to quickly and easily yield good classification accuracy—for example in example speech recognition

or hand-written digit identification (Hastie et al., 2009). Beyond accuracy, however, neuroscientists often aim to understand which neural features are related to particular stimuli or behaviors at specific points in time. This distinct aim of interpretability requires classification or regression methods that can yield clearly interpretable sets of model coefficients. For this reason, the recent literature on classification of fMRI data has recommended using linear classifiers (e.g., logistic regression (LR), linear discriminant analysis (LDA), Gaussian Naive Bayes (GNB), or linear SVM) rather than nonlinear classifiers (Haynes and Rees, 2006; Pereira et al., 2009).

Linearity alone, however, does not guarantee that a method will yield a stable and interpretable solution. For instance, in the case of multiple correlated input variables LR, LDA, and GNB yield unstable coefficients and degenerate covariance estimates, particularly when applied to smoothed data (Hastie et al., 1995, 2009). In the context of classification, penalized least squares may over smooth coefficients, complicating interpretation (Friedman, 1997). Additionally, most linear classifiers return dense sets of coefficients (as in Fig. 1, left panels) that require subsequent thresholding or feature selection to yield parsimonious solutions. Although heuristic methods exist for coefficient selection, these are generally greedy (e.g., forward/backward stage-wise procedures like Recursive Feature Elimination (Bray et al., 2009; De Martino et al., 2007; Guyon et al., 2002)), yielding unstable solutions when data are resampled (since these algorithms tend to converge to local minima) (Hastie et al., 2009). Although principled methods exist for applying thresholds to dense mass-univariate coefficient maps (e.g. Random Field Theory (Adler and Taylor, 2000; 122

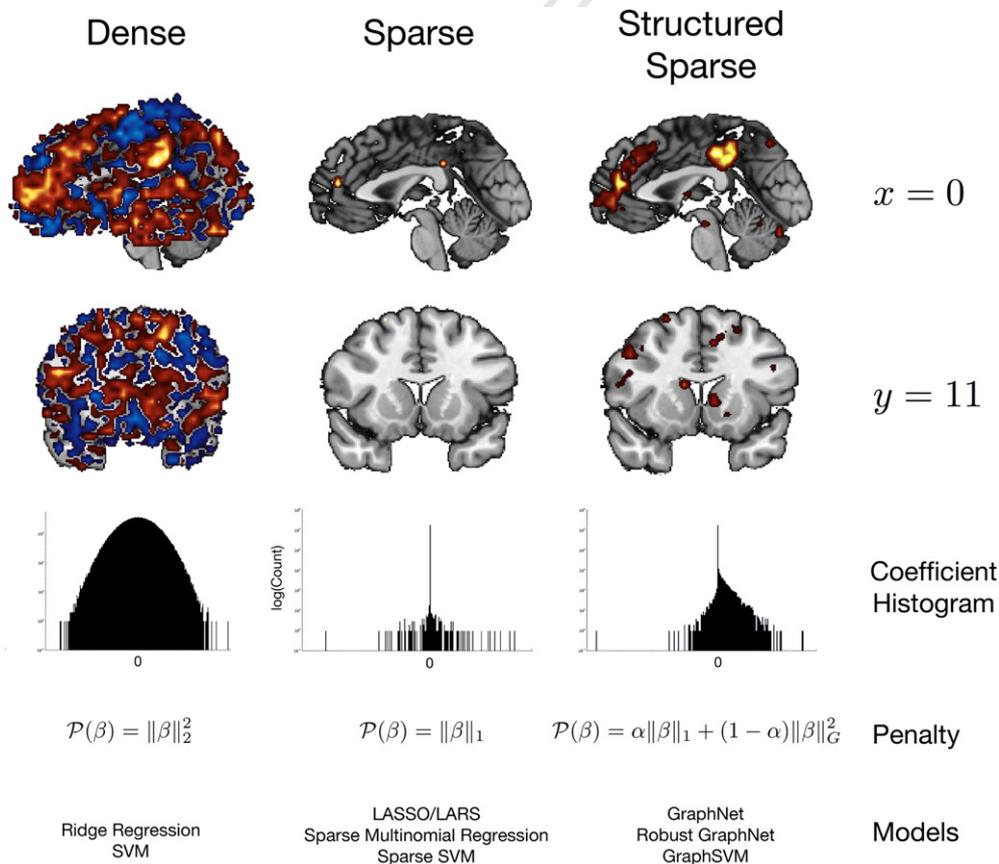


Fig. 1. Mid-sagittal and coronal plots of example coefficients from dense, sparse, and structured sparse coefficients (in Talairach coordinates). Warm colored coefficients indicate a positive relationship with the target variable (here predicting the decision to buy a product), cool colors a negative relationship. Sparse methods set many coefficients to zero, while in dense methods almost all coefficients are nonzero. Structured sparse methods use a penalty on differences between selected voxels to impose a structure on the fit so that it yields coefficients that are both sparse and structured (e.g., smooth). Log-histograms of the estimated voxel-wise coefficients show that the sparse method coefficients have a near-Laplacian (double-exponential) distribution, while the dense coefficients have a near-Gaussian distribution. The structured sparse coefficients are a product of these distributions (see also Fig. 2). Coefficient penalties that yield each result and examples of related methods are given below each column.

Worsley et al., 2004)), these approaches do not currently extend to dense multivariate regression or classification methods.

Recently, sparse regression methods have been applied to neuroimaging data to yield reduced coefficient sets that are automatically selected during model fitting. The first examples in the fMRI literature include Yamashita et al. (2008), who applied sparse logistic regression (Tibshirani, 1996) to classification of visual stimuli, and Grosenick et al. (2008) who first developed sparse penalized discriminant analysis by converting an “Elastic Net” regression (Zou and Hastie, 2005) into a classifier, and then applied it to choice prediction. Subsequently, sparse methods for regression (Carroll et al., 2009; Hanke et al., 2009) and classification (Hanke et al., 2009) have been applied to fMRI data to yield reduced sets of coefficients from volumes of interest, whole-brain volumes (Ryali et al., 2010; van Gerven et al., 2010), and whole-brain volumes over multiple time points (Grosenick et al., 2009b, 2010). These methods typically impose an ℓ_1 -penalty (sum of absolute values) on the model coefficients, which sets many of the estimated coefficients to zero (see Fig. 1, leftmost panels, and Fig. 2b). When applied to correlated fMRI data, however, ℓ_1 -penalized methods can select an overly sparse solution—resulting in omission of relevant features as well as unstable coefficient estimates during cross-validation (Grosenick et al., 2008; Zou and Hastie, 2005). To allow relevant but correlated coefficients to coexist in a sparse model fit, recent approaches to fMRI regression (Carroll et al., 2009; Li et al., 2009) and classification (Grosenick et al., 2008, 2009b; Ryali et al., 2010) impose a hybrid of both ℓ_1 - and ℓ_2 -norm penalties (the “Elastic Net” penalty of Zou and Hastie (2005) on the coefficients. These hybrid approaches allow the inclusion of correlated variables in sparse model fits.

This paper explores modified methods that combine the Elastic Net penalty with a general user-specified sparse graph penalty. This sparse graph penalty allows the user to efficiently incorporate physiological constraints and prior information (such as smoothness in space or time or anatomical details such as topology or connectivity) in the model. The resulting graph-constrained elastic net (or “GraphNet”) regression (Grosenick et al., 2009b, 2010) has the capacity to find “structured sparsity” in correlated data with many features (Fig. 1, right panels), consistent with results in the manifold learning (Belkin et al., 2006) and gene microarray literatures (Li and Li, 2008). In the statistics literature, related “sparse structured” methods have been shown to have desirable convergence and variable selection properties for large correlated data sets (Jenatton et al., 2011; Slawski et al., 2010).

These sparse, structured models can also be implemented within a Bayesian framework (van Gerven et al., 2010). Here, we extend the performance of GraphNet regression and classification methods to whole-brain fMRI data by: (1) generalizing them to be robust to outliers in fMRI data (for both regression and classification), (2) adding “adaptive” penalization to reduce fit bias and improve variable selection, and (3) developing a novel support vector GraphNet (SVGN) classifier. Additionally, to efficiently fit GraphNet methods to whole-brain fMRI data over multiple time-points, we adapt algorithms from the applied statistics literature (Friedman et al., 2010).

After developing robust and adaptive GraphNet regression and classification methods, we demonstrate the enhanced performance of GraphNet classifiers on previously published data (Karmarkar et al., submitted for publication; Knutson et al., 2007). Specifically, we use GraphNet methods to predict subjects’ trial-to-trial purchasing behavior with whole-brain data over several time points, and then infer which brain regions best predict upcoming choices to purchase or not purchase a product. Fitting these methods to 25 subjects’ whole-brain data over 7 time points (2 s TRs) yielded classification rates which exceeded those found previously in a volume of interest (VOI) based classification analysis (Grosenick et al., 2008), as well as those obtained with a linear support vector machine (SVM) classifier fit to the whole brain data. While the GraphNet results on whole-brain data confirm the relevance of previously chosen volumes of interest (i.e., bilateral nucleus accumbens (NAcc), medial prefrontal cortex (MPFC), and anterior insula), they also implicate previously unchosen areas (i.e., ventral tegmental area (VTA) and posterior cingulate). We conclude with a discussion of the interpretation of GraphNet model coefficients, as well as future improvements, applications, and extensions of this family of GraphNet methods to neuroimaging data. Open source code for solving the GraphNet problems in this paper is freely available at <https://github.com/logang/neuroparser>.

Methods

Background

Penalized least squares

Many classification and regression problems can be formulated as modeling a response vector $y \in \mathbb{R}^n$ as a function of data matrix $X \in \mathbb{R}^{n \times p}$, which consists of n observations each of length p (with $n \geq p$).

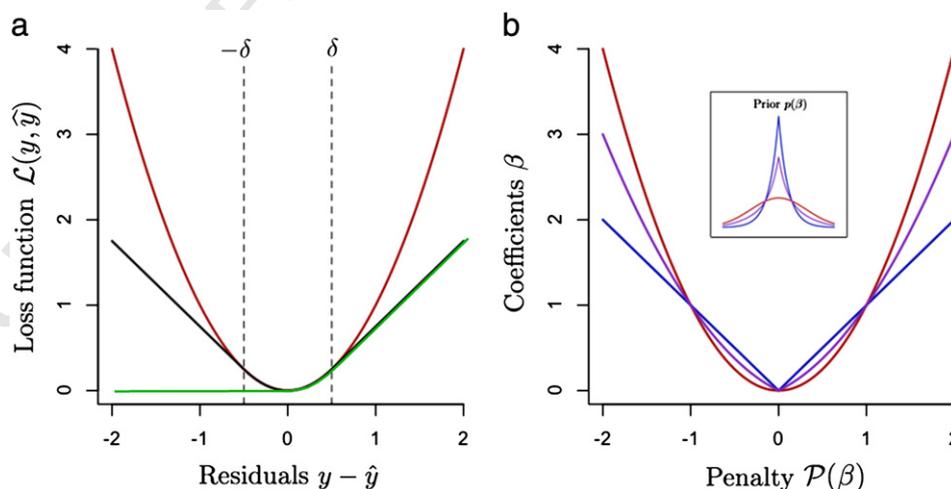


Fig. 2. (a) Diagrammatic representation of squared-error (red), Huber (black), and Huberized Hinge (green) loss functions. Dotted lines denote where the Huber loss changes from penalizing residuals quadratically (where $|y - \hat{y}| \leq \delta$) to penalizing them linearly (where $|y - \hat{y}| > \delta$). The linear penalty on large residuals makes the Huber loss robust. (b) Diagrammatic representation of convex penalty functions used in this article (along one coordinate β). The red curve is a quadratic penalty $\mathcal{P}(\beta) = \beta^2$ on coefficient magnitude, often called the Tikhonov or “ridge” penalty in regression. The blue curve is the Lasso penalty on coefficient magnitude $\mathcal{P}(\beta) = |\beta|$. The purple curve is a convex combination of the red and blue curves: $\mathcal{P}(\beta) = \alpha\beta^2 + (1 - \alpha)|\beta|$ (where here $\alpha = 0.5$, called the “Elastic Net” penalty. The inset shows the shape of the prior distribution on the coefficient estimates that each of these penalties corresponds to: Gaussian (red), Laplacian (blue), and mixed Gaussian and Laplacian (purple) (units arbitrary). The priors become increasingly peaked around zero as the Elastic Net penalty approaches the Lasso penalty, corresponding to a prior belief that many coefficients will be exactly zero.

In particular, a large number of models treat y as a linear combination of the predictors in the presence of noise $\epsilon \in \mathbb{R}^n$, such that

$$y = X\beta + \epsilon, \quad (1)$$

where ϵ is a noise vector typically assumed to be normally distributed $\epsilon \sim N(0, I\sigma^2)$ with vector mean 0 and diagonal variance–covariance matrix $I\sigma^2$, and $\beta \in \mathbb{R}^p$ a vector of linear model coefficients. In this case using squared error loss leads to the well-known ordinary least squares (OLS) solution

$$\hat{\beta} = \arg \min_{\beta} \|y - X\beta\|_2^2 = (X^T X)^{-1} X^T y, \quad (2)$$

which yields the best linear unbiased estimator (BLUE) if the columns of X are uncorrelated (Lehmann and Casella, 1998).

However, this estimator is inefficient in general for $p > 2$ —it is dominated by biased estimators (Stein, 1956)—and if the columns of X are correlated (i.e. are “multicollinear”) then the estimated coefficient values can vary erratically with small changes in the data, so the OLS fit can be quite poor. A common solution to this problem is penalized (or “regularized”) least squares regression (Tikhonov, 1943), in which the magnitude of the model coefficients are penalized to stabilize them. This is accomplished by adding a penalty term $\mathcal{P}(\beta)$ on the coefficient vector β , yielding

$$\hat{\beta} = \arg \min_{\beta} \|y - X\beta\|_2^2 + \lambda \mathcal{P}(\beta), \lambda \in \mathbb{R}_+, \quad (3)$$

where λ is a parameter that trades off least squares goodness-of-fit with the penalty on the model coefficients (or equivalently, trades off fit variance for fit bias) and \mathbb{R}_+ is the set of nonnegative scalars. These estimates are equivalent to maximum a posteriori (MAP) estimates from a Bayesian perspective (with a Gaussian prior on the coefficients if $\mathcal{P}(\beta) = \|\beta\|_2^2$ (Hastie et al., 2009)), or to the Lagrangian relaxation of a constrained bi-criterion optimization problem (Boyd and Vandenberghe, 2004). Such equivalencies motivate various interpretations of the model coefficients and parameter λ (see the section “Interpreting GraphNet regression and classification”).

Sparse regression and automatic variable selection

There are a few standard choices for the penalty $\mathcal{P}(\beta)$. Letting $\mathcal{P}(\beta) = \|\beta\|_2^2 = \sum_{j=1}^p \beta_j^2$ (the ℓ_2 norm) gives the classical Tikhonov or “ridge” regression estimates originally proposed for such problems (Hoerl and Kennard, 1970; Tikhonov, 1943). More recently, the choice $\mathcal{P}(\beta) = \|\beta\|_1 = \sum_{j=1}^p |\beta_j|$ (the ℓ_1 norm)—called the Least Absolute Shrinkage and Selection Operator (or “Lasso”) penalty in the regression context (Tibshirani, 1996)—has become widely popular in statistics, engineering, and computer science, leading some to call such ℓ_1 -regression the “modern least squares” (Candes et al., 2008). In addition to shrinking the coefficient estimates, the Lasso performs variable selection by producing sparse coefficient estimates (i.e., many are exactly equal to zero, see Fig. 1 left panels). In many applications, having a sparse vector $\hat{\beta}$ is highly desirable, since a fit with fewer non-zero coefficients is simpler, and can help select predictors that have an important relationship with the response variable y .

The ℓ_1 -norm used in the Lasso is the closest convex relaxation of the ℓ_0 pseudo-norm $\|\beta\|_0 = \sum_{j=1}^p 1_{\{\beta_j \neq 0\}}$, where $1_{\{\beta_j \neq 0\}}$ is an indicator function that is 1 if the j th coefficient β_j is nonzero and 0 otherwise. This represents a penalty on the number of nonzero coefficients (their cardinality). However, finding a minimal cardinality solution generally involves a combinatorial search through possible sets of nonzero coefficients (a form of “all subsets regression” (Hastie et al., 2009)) and so is computationally infeasible for even a modest number of input features. An ℓ_1 -norm penalty can be used as a heuristic that results in coefficient sparsity (which corresponds to the maximum a posteriori (MAP) estimates under a Laplacian (double-exponential) prior; for a fully Bayesian

approach see van Gerven et al. (2010)). Such ℓ_1 -penalized regression methods set many variables equal to zero and automatically select only a small subset of relevant variables to assign nonzero coefficients. While these methods yield the sparsest possible fit in many cases (Candes et al., 2003; Donoho, 2006), they do not always do so, and reweighted methods (e.g., Automatic Relevance Determination (ARD) (Wipf and Nagarajan, 2008) and iterative reweighting of the ℓ_1 penalty (Candes et al., 2008)) exist for finding sparser solutions. It is worth noting that while Bayesian methods for variable selection (such as Relevance Vector Machines) have existed in the literature for some time, these methods typically require using EM-like or MCMC approaches that do not guarantee convergence to a global minimum and that are relatively computationally inefficient (though see Mohamed et al. (2011) for an interesting counter-point). As an interesting exception, recent work on ARD and sparse Bayesian learning (Wipf and Nagarajan, 2008) have provided an attractive alternative, showing that the sparse Bayesian learning problem can be solved as a sequence of reweighted Lasso problems, similar to the adaptive methods discussed below. This approach no longer provides a full posterior, but does provide an interesting and computationally tractable link to the Bayesian formulation. In the future we expect that such links will lead to better approaches for model selection in these methods than the “brute force” grid search employed here.

Elastic Net regression

Despite offering a sparse solution and automatic variable selection, there are several disadvantages to using ℓ_1 -penalized methods like the Lasso in practice. For example, from a group of highly correlated predictors, the Lasso will typically select a subset of “representative” predictors to include in the model fit (Zou and Hastie, 2005). This can make it difficult to interpret coefficients because those that are set to 0 may still be useful for modeling y (i.e., false negatives are likely). Worse, entirely different subsets of coefficients may be selected when the data are resampled (e.g., during cross-validation). Moreover, the Lasso can select at most n non-zero coefficients (Zou and Hastie, 2005), which may prove undesirable when the number of input features (p) exceeds the number of observations (n) (i.e., “ $p \gg n$ ” problems). Finally, as a global shrinkage method, the Lasso biases model coefficients towards zero (Hastie et al., 2009; Tibshirani, 1996), making interpretation with respect to original data units difficult. Other methods that use only an ℓ_1 penalty (e.g., sparse logistic/multinomial regression and sparse SVM (Hastie et al., 2009)) are subject to the same deficiencies.

In response to several of these concerns Zou and Hastie (2005) proposed the Elastic Net, which uses a mixture of ℓ_1 - and ℓ_2 -norm regularization, and may be written

$$\hat{\beta} = \kappa \arg \min_{\beta} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2, \quad (4)$$

where the factor $\kappa = 1 + \lambda_2$ in (4) and subsequent equations is a rescaling factor discussed in further detail below. This Elastic Net estimator overcomes several (though not all) of the disadvantages discussed above, while maintaining many advantages of Tikhonov (“Ridge”) regression and the Lasso. In particular, the Elastic Net accommodates groups of correlated variables and can select up to p variables with non-zero coefficients. The amount of sparsity in the solution vector can be tuned by adjusting the penalty coefficients λ_1 and λ_2 . In this case, the ℓ_1 penalty can be thought of as a heuristic for enforcing sparsity, while the ℓ_2 penalty allows correlated variables to enter the model and stabilizes the sample covariance estimate. This Elastic Net approach performs well on fMRI data in both regression and classification settings (Carroll et al., 2009; Grosenick et al., 2008; Ryali et al., 2010).

Graph-constrained Elastic Net (GraphNet) regression

So far we have seen that sparse regression methods like the Elastic Net, which use a hybrid ℓ_1 - and ℓ_2 -norm penalty, can be used to yield

321 sparse model fits that do not exclude correlated variables (Zou and
 322 Hastie, 2005), and that we can turn these regression methods into
 323 classifiers that perform well when fit to VOI data (Grosenick et al.,
 324 2008). However, the Elastic Net penalty merely makes the model
 325 fitting procedure “blind” to correlations between input features (by
 326 shrinking the sample estimate of the covariance matrix towards the
 327 identity matrix). Indeed, if λ_2 in Eq. (9) grows large, this method is
 328 equivalent to applying a threshold to mass-univariate OLS regression
 329 coefficients (i.e., the estimate of the covariance matrix becomes a
 330 scaled identity matrix) (Zou and Hastie, 2005).

331 In this section, we describe a modification of the Elastic Net that
 332 explicitly imposes structure on the model coefficients. This allows
 333 the analyst to pre-specify constraints on the model coefficients (e.g.,
 334 based on prior information like local smoothness or connectivity, or
 335 other desirable fit properties), and then to tune how strongly the fit
 336 adheres to these constraints. Since the user-specified constraints
 337 take the general form of an undirected graph, we call this regression
 338 method the graph-constrained Elastic Net (or “GraphNet”) (Grosenick
 339 et al., 2009b, 2010).

340 The GraphNet model closely resembles the Elastic Net model, but
 341 with a modification to the ℓ_2 -norm penalty term:

$$\hat{\beta} = \kappa \arg \min_{\beta} \|y - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_G \|\beta\|_G^2 \quad (5)$$

$$\|\beta\|_G^2 = \beta^T G \beta = \sum_{j=1}^p \sum_{k=1}^p \beta_j G_{jk} \beta_k,$$

342 where G is a sparse graph. Note that in the case where $G = I$, where I
 343 denotes the identity matrix, the GraphNet reduces back to the Elastic
 344 Net. Thus the Elastic Net is a special case of GraphNet and we can repli-
 345 cate the effects of increasing an Elastic Net penalty by adding a
 346 scaled version of the identity matrix (λ_2/λ_G) I to G (for $\lambda_G > 0$).

347 The example we will use for the matrix G in the remainder of this
 348 paper is the graph Laplacian, which formalized our intuition that
 349 voxels that are neighbors in time and space should typically have
 350 similar values. If we take the coefficients β to be functions over the
 351 brain volume $V \in \mathbb{R}^3$ over time points $T \in \mathbb{R}$ such that $\beta(x, y, z, t)$,
 352 then we would like a penalty that penalizes roughness in the coeffi-
 353 cients as measured by their derivatives over space and time, such as

$$\mathcal{P}(\beta) = \int_{V,T} \left(\frac{\partial^2 \beta}{\partial x^2} + \frac{\partial^2 \beta}{\partial y^2} + \frac{\partial^2 \beta}{\partial z^2} + \frac{\partial^2 \beta}{\partial t^2} \right) dx dy dz dt$$

$$= \int_{V,T} \Delta \beta dx dy dz dt, \quad (6)$$

354 where Δ is the Laplacian operator, which here is a 4D isotropic measure
 355 of the second spatio-temporal derivative of the volumetric time-series.
 356 Since we are sampling discretely, we use the discrete approximation to
 357 the Laplacian operator Δ : the matrix Laplacian $L = D - A$ (the difference
 358 between the degree matrix D and the adjacency matrix A , see e.g.,
 359 (Hastie et al., 1995)). This formulation generalizes well to arbitrary
 360 graph connectivity and is widely used in spectral clustering techniques
 361 and spectral graph theory (Belkin and Niyogi, 2008).

362 In the case where $G = L$, the graph penalty, $\|\beta\|_G^2$, has the appeal-
 363 ingly simple representation

$$\|\beta\|_G^2 = \sum_{(i,j) \in \varepsilon_G} (\beta_i - \beta_j)^2,$$

364 where ε_G is the set of index pairs for voxels that share an edge in graph
 365 G (i.e. have a nonzero entry in the adjacency matrix A). Written this
 366 way, the graph penalty induces smoothness by penalizing the size of
 367 the pairwise differences between coefficients that are adjacent in the
 368 graph. In the one dimensional case, if the quadratic terms $(\beta_i - \beta_j)^2$
 369 were replaced by absolute deviations $|\beta_i - \beta_j|$ then this would in-
 370 stead be an instance of the “fused Lasso” (Tibshirani et al., 2005) or

Generalized Lasso (Tibshirani and Taylor, 2011). There are two main
 reasons for preferring a quadratic penalty in the present application:

1. The fused Lasso is closely related to Total Variation (TV) denoising (Rudin et al., 1992) and tends to set many of the pairwise differences $\beta_i - \beta_j$ to zero, creating a sharp piecewise constant set of coefficients that lacks the spatial smoothness often expected in fMRI data. Extending this formulation to processes with more than one spatial or temporal dimension is nontrivial (Michel et al., 2011).
2. Significant algorithmic complications can be avoided by choosing a differentiable penalty on the pairwise differences (Friedman et al., 2007a; Tseng, 2001), speeding up model fitting and reducing model complexity considerably—especially in the case of spatial data, where the Total Variation penalty must be formulated as a more complicated sum of non-smooth norms on each of the first-order forward finite difference matrices (Michel et al., 2011; Wang et al., 2008b).

Thus GraphNet methods provide a sparse and structured solution similar to the Fused Lasso, Generalized Lasso, and Total Variation. However, unlike these approaches, GraphNet methods allow for smooth rather than piecewise constant structure in the non-sparse parts of the reconstructed volume. This is of interest in cases where we might expect the magnitude of nonzero coefficients to be different within a volume of interest. Due to the smoothness of the graph penalty GraphNet methods are also easier from an optimization perspective. Of course, there are certainly situations in which the piecewise smoothness of Total Variation could be a better prior (this depends on the data and problem formulation).

Adaptive GraphNet regression

The methods described above automatically select variables by shrinking coefficient estimates towards zero, resulting in downwardly biased coefficient magnitudes. This shrinkage makes it difficult to interpret coefficient magnitude in terms of original data units, and severely restricts the conditions under which the Lasso can perform consistent variable selection (Zou, 2006). Ideally, given infinite data, the method would select the correct parsimonious set of features (i.e., the “true model”, were it known), but avoid shrinking nonzero coefficients that remain in the model (unbiased estimation). Together, these desiderata are known as the “oracle” property (Fan and Li, 2001). Note that in the neuroimaging context, the first (consistent variable selection) corresponds to correct localization of signal, while the second (consistent coefficient estimation) relates to improving estimates of coefficient magnitude.

Several estimators possessing the oracle property (given certain conditions on the data) have been reported in the literature, including the Adaptive Lasso (Zhou et al., 2011; Zou, 2006) and the Adaptive Elastic Net (Zou and Zhang, 2009). These estimators are straightforward modifications of penalized linear models. They work by starting with some initial estimates of the coefficients obtained by fitting the non-adaptive model (Zou and Zhang, 2009), and use these to adaptively reweight the penalty on each individual coefficient β_j , $j = 1, \dots, p$. Recently (Slawski et al., 2010) extended the adaptive approach to a sparse, structure method equivalent to GraphNet regression, and proved that the oracle properties previously shown for the adaptive Lasso and Adaptive Elastic Net extend to the sparse, structured case provided the true coefficients are in the null space of G (i.e. the nonzero entries of β specify a connected component in G). We refer the reader to (Slawski et al., 2010) for further details.

As in (Slawski et al., 2010), we may rewrite the GraphNet to have an adaptive penalty (the adaptive GraphNet) as follows:

$$\hat{\beta} = \arg \min_{\beta} \|y - X\beta\|_2^2 + \lambda_1 \sum_{j=1}^p \hat{w}_j |\beta_j| + \lambda_G \|\beta\|_G^2 \quad (7)$$

$$\hat{w}_j = |\tilde{\beta}_j|^{-\gamma}. \quad (8)$$

The idea here is that important coefficients will have large starting estimates $\tilde{\beta}_j$ (where β_j is a suitable estimator of β_j) and so will be shrunk at a rate inversely proportional to their starting estimates, leaving them asymptotically unbiased. On the other hand, coefficients with small starting estimates $\tilde{\beta}_j$ will experience additional shrinkage, making them more likely to be excluded. We let $\gamma = 1$ as in the finite sample case (Zou, 2006; Zou and Zhang, 2009), and by analogy to the Adaptive Elastic Net (Zou and Zhang, 2009) set $\tilde{\beta}$ to the standard GraphNet coefficient estimates for a fixed value of λ_G (chosen based on the GraphNet performance at that value). We use λ_G^* to differentiate the adaptive fit sparsity parameter from the parameter associated with the GraphNet fit used to initialize the weights \hat{w}_j .

It is important to note that the oracle properties that hold in the asymptotic case may not apply to the finite sample, $p \gg n$ situation. Nevertheless, we include these methods for comparison since oracle properties are desirable and since evidence suggests that the Adaptive Elastic Net has improved finite sample performance because it deals well with collinearity (Zou and Zhang, 2009).

Turning sparse regression methods into classifiers: Optimal Scoring (OS) and Sparse Penalized Discriminant Analysis (SPDA)

Sparse regression methods like the Lasso or Elastic Net can be turned into sparse classifiers (Clemmensen et al., 2011; Grosenick et al., 2008; Leng, 2008). Naively, we might imagine performing a two-class classification simply by running a regression with Lasso or the Elastic Net on a target vector containing 1's and 0's depending on the class of each observation $y_i \in \{0, 1\}$. We would then take the predicted values from the regression \hat{y} and classify to 0 if the i th estimate $\hat{y}_i < 0.5$ and to 1 if the estimate $\hat{y}_i > 0.5$ (for example). In the multi-class case (i.e. J classes with $J > 2$), multi-response linear regression could be used as a classifier in a similar way. This would be done by constructing an indicator response matrix Y , with n rows and J columns (where again n is the number of observations and J is the number of classes). Then the i th row of Y has a 1 in the j th column if the observation is in the j th class and a 0 otherwise. If we run a multiple linear regression of Y on predictors X , we can classify by assigning the i th observation to the class having the largest fitted value $\hat{Y}_{i1}, \hat{Y}_{i2}, \dots, \hat{Y}_{iJ}$. With the exception of binary classification on balanced data, this classifier has several disadvantages. For instance, the estimates \hat{Y}_{ij} are not probabilities, and in the multi-class case certain classes can be "masked" by others, resulting in decreased classification accuracy (Hastie et al., 2009). However, applying LDA to the fitted values of such a multiple linear regression classifier is mathematically equivalent to fitting the full LDA model (Breiman and Ihaka, 1984), yielding posterior probabilities for the classes and dramatically improving classifier performance over the original multivariate regression in some cases (Hastie et al., 1994, 1995, 2009).

Hastie et al. (1994) and Hastie et al. (1995) exploit equivalences between multiple regression and LDA and between LDA and canonical correlation analysis to develop a procedure they call Optimal Scoring (OS). OS allows us to build a classifier by first fitting a multiple regression to Y using an arbitrary regression method, and then linearly transforming the fitted results of this regression using the OS procedure (see Hastie et al. (1994) for further algorithmic and statistical details). This procedure yields both class probability estimates and discriminant coordinates, and allows us to use any number of regression methods as discriminant classifiers. This approach is discussed in detail for nonlinear regression methods applied to a few input features in Hastie et al. (1994), and for regularized regression methods applied to numerous (i.e., hundreds) of correlated input features in Hastie et al. (1995). Here we extend the results of the latter work to include sparse structured regression methods that can be fit efficiently to hundreds of thousands of input features.

More formally, OS finds an optimal scoring function $\theta: g \rightarrow R$ that maps classes $g \in \{1, \dots, J\}$ into the real numbers. In the case of a multi-class classification using the Elastic Net, we can apply OS to yield estimates

$$(\hat{\theta}, \hat{\beta}) = \kappa \arg \min_{\theta, \beta} \|Y\theta - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2 \quad (9)$$

$$\text{subject to } n^{-1} \|Y\theta\|_2^2 = 1, \quad (10)$$

where θ is a matrix that yields the optimal scores when applied to indicator matrix Y , and where we add the constraint (9) to avoid degenerate solutions (Grosenick et al., 2008). Given that this is just a sparse version of PDA (Hastie et al., 1995), we have called this combination Sparse Penalized Discriminant Analysis (SPDA). It has also recently been called Sparse Discriminant Analysis (SDA) (Clemmensen et al., 2011) (and for an interesting alternative approach for constructing sparse linear discriminant classifiers, see Witten and Tibshirani (2011)).

For simplicity, we consider only a local spatiotemporal smoothing penalty in the current study, although using more elaborate spatial/temporal coordinates would follow similar logic. The SPDA-GraphNet is defined as

$$(\hat{\theta}, \hat{\beta}) = \kappa \arg \min_{\theta, \beta} \|Y\theta - X\beta\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_G \|\beta\|_2^2 \quad (11)$$

$$\text{subject to } n^{-1} \|Y\theta\|_2^2 = 1. \quad (12)$$

It is important to note that the direct equivalence between penalized OS and penalized LDA has only recently been proven in the binary classification case, and does not hold for multi-class classification problems (Merchante et al., 2012). However, both approximate methods that iteratively minimize over θ and β (Clemmensen et al., 2011) and equivalent methods based on the Group Lasso (Merchante et al., 2012) could be used with GraphNet regression methods to build multi-class GraphNet classifiers. We note that in the binary classification case there are at least two options to turn regression methods into classifiers: Optimal Scoring and logistic regression (see e.g. Friedman et al., 2010). In the case of multiple classes, the approaches of (Clemmensen et al., 2011; Merchante et al., 2012) provide LDA or LDA-like classifiers. Sparse multinomial regression could also be used in the multi-class case. Any of these approaches may be used to turn GraphNet regression methods into GraphNet classifiers. Because Optimal Scoring converts regression methods into equivalent linear discriminant classifiers, it allows us to combine notions from regression such as degrees of freedom with notions from discriminant analysis such as class visualization in the discriminant space using discriminant coordinates and trial-by-trial posterior probabilities for individual observations (Hastie et al., 1995). This, and its greater computational simplicity over logistic and multinomial regressions, make OS an appealing approach.

Turning regression methods into classifiers: relating Support Vector Machines (SVM) to penalized regression

In addition to the LDA and logistic/multinomial approaches to classification, maximum margin classifiers like SVM have been very successful. As we will also be developing a Support Vector GraphNet (SVGN) variant below, we briefly discuss how support vector machines can be related to regression methods like those described above. If the data is centered such that an intercept term can be ignored, the SVM solution can be written

$$\hat{\beta} = \arg \min_{\beta} \sum_{i=1}^n (1 - y_i x_i^T \beta)_+ + (\lambda/2) \|\beta\|_2^2,$$

556 where $(\cdot)_+$ indicates taking the positive part of the quantity in paren-
 557 theses. In this function estimation formulation of the SVM problem,
 558 we see the similarity to the penalized regression methods above:
 559 the only difference is that the usual squared error loss $L(y_i, x_i, \beta) =$
 560 $(y_i - x_i^T \beta)^2$ has been replaced by the “hinge loss” function $L_H(y_i,$
 561 $x_i, \beta) = (1 - y_i x_i^T \beta)_+$. This function is non-differentiable, and more re-
 562 cent work (Wang et al., 2008a) uses a differentiable “Huberized
 563 hinge loss” (Fig. 2a), which we will discuss in greater detail below.
 564 The important point here is that formulating the SVM problem as a
 565 loss term and a penalty term reveals how we might build an SVM
 566 with more general penalization, such as that used in GraphNet re-
 567 gression methods above.

568 *Novel extensions of GraphNet methods*

569 *Robust GraphNet and Adaptive Robust GraphNet*

570 More generally, we can formulate the penalized regression prob-
 571 lem of interest as minimizing the penalized empirical risk $R_p(\beta)$ as a
 572 function of the coefficients, so that

$$\hat{\beta} = \arg \min_{\beta} \mathcal{R}_p(\beta) = \arg \min_{\beta} \mathcal{R}(y, \hat{y}) + \lambda \mathcal{P}(\beta), \quad (13)$$

574 where \hat{y} is the estimate of response variable y (note $\hat{y} = X\hat{\beta}$ in the lin-
 575 ear models we consider) and $R(y, \hat{y}) = n^{-1} \sum_{i=1}^n L(y_i, \hat{y}_i)$ is the average
 576 of the loss function over the training data (the “empirical risk”) of the
 577 loss function $L(y_i, \hat{y}_i)$ that penalizes differences between the estimat-
 578 ed and true values of y at the i th observation. For example, in Eqs.
 579 (3)–(9) we used $\mathcal{R}(y, \hat{y}) = \|y - \hat{y}\|_2^2 = \sum_{i=1}^n (y_i - \hat{y}_i)^2$ (“squared error
 580 loss”). While squared error loss enjoys many desirable properties
 581 under the assumption of Gaussian noise, it is sensitive to the presence
 582 of outliers.

583 Outlying data points are an important consideration when modeling
 584 fMRI data, in which a variety of factors ranging from residual motion ar-
 585 tifacts to field inhomogeneities can cause some observations to fall far
 586 from the sample mean. In the case of standard squared-error loss (as
 587 in Eqs. (2)–(9)), these outliers can have undue influence on the model
 588 fit due to the quadratically increasing penalty on the residuals (see
 589 Fig. 2a). A standard solution in such cases is to use a robust loss function,
 590 such as the Huber loss function (Huber and Ronchetti, 2009),

$$\mathcal{R}_H(y, \hat{y}; \delta) = n^{-1} \sum_{i=1}^n L_{\delta}(y_i - \hat{y}_i) \quad (14)$$

592 where

$$L_{\delta}(y_i - \hat{y}_i) = \begin{cases} (y_i - \hat{y}_i)^2 / 2 & \text{for } |y_i - \hat{y}_i| \leq \delta \\ \delta |y_i - \hat{y}_i| - \delta^2 / 2 & \text{for } |y_i - \hat{y}_i| > \delta \end{cases}$$

594 This function penalizes residuals quadratically when they are less
 595 than or equal to parameter δ , and linearly when they are larger than δ
 596 (Fig. 2a). A well specified δ can thus significantly reduce the effects of
 597 large residuals (outliers) on the model fit, as they no longer have the
 598 leverage resulting from a quadratic penalty. As $\delta \rightarrow \infty$ (or practically,
 599 when it becomes larger than the most outlying residual) we recover
 600 the standard squared-error loss.

602 Since GraphNet uses squared-error loss, it can now be modified to in-
 603 clude a robust penalty like the Huber loss defined above. Replacing the
 604 squared error loss function with the loss function (Eq. (14)) yields

$$\hat{\beta} = \kappa \arg \min_{\beta} \mathcal{R}_H(y, X\beta; \delta) + \lambda_1 \|\beta\|_1 + \lambda_G \|\beta\|_G^2. \quad (15)$$

606 The Adaptive Robust GraphNet is then a straightforward generali-
 607 zation (following the section “Adaptive GraphNet regression”; see also
 608 the next section).
 609

The SPDA-RGN classifier can be defined like the standard GraphNet
 classifier (Eq. (11)). However, the SPDA-RGN classifier now has an ad-
 ditional hyperparameter to be estimated (or assumed). Specifically,
 the value of δ determines where the loss function switches from qua-
 dratic to linear (Fig. 2a). Further, the loss function on the residuals is
 no longer quadratic and therefore could slow down optimization con-
 vergence. We discuss a solution to this issue next.

Infimal convolution for non-quadratic loss functions

In order to solve both the Robust GraphNet, Adaptive Robust
 GraphNet, and Support Vector GraphNet problems efficiently, we in-
 troduce a general method for solving coordinate-wise descent prob-
 lems with smooth, non-quadratic convex loss functions as penalized
 least squares problems in an augmented set of variables.

Convergence speed of subgradient methods such as coordinate-
 wise descent can be substantially improved when the loss function
 takes a quadratic form, while non-quadratic loss functions can take
 numerous iterations to converge for each coefficient, significantly in-
 creasing computation time. However, we can circumvent these prob-
 lems and extend the applicability of coordinate-wise descent methods
 using a trick from convex analysis to rewrite these loss functions as
 quadratic forms in an augmented set of variables. This method is
 called infimal convolution (Rockafellar, 1970), and is defined as

$$(f \star_{\text{inf}} g)(x) := \inf_y \{f(x-y) + g(y) | y \in \mathbb{R}^n\}, \quad (16)$$

where f and g are two functions of $x \in \mathbb{R}^p$. In this way it is possible to
 rewrite the i th term in the Huber loss function (Eq. (14)) as the infimal
 convolution of the squared and absolute-value functions applied to
 the i th residual r_i :

$$\rho_{\delta}(r_i) = \left((1/2) (\cdot)^2 \star_{\text{inf}} |\cdot| \right) (r_i) = \inf_{a_i + b_i = r_i} a_i^2 / 2 + \delta |b_i|, \quad (17)$$

where $r_i = y_i - (X\hat{\beta})_i$ (note that a dot (\cdot) is used to indicate the func-
 tional nature of the expression without having to add additional
 dummy variables). This yields the augmented estimation problem

$$\begin{aligned} (\hat{\alpha}, \hat{\beta}) = \arg \min_{\alpha, \beta} & (1/2) \|y - X\beta - \alpha\|_2^2 + \lambda_G \beta^T G \beta + \delta \|\alpha\|_1 \\ & + \lambda_1 \|\beta\|_1, \end{aligned} \quad (18)$$

where we have introduced the auxiliary variables $\alpha \in \mathbb{R}^n$. Considering
 the residuals r_i , the first term in the objective of Eq. (18) can be written
 $(1/2) \|y - X\beta - \alpha\|_2^2 = (1/2) \sum_i (r_i - \alpha_i)^2$ and thus each α_i can directly
 reduce the residual sum of squares corresponding to a single observa-
 tion by taking a value close to r_i . Since for some δ the penalty $\delta \|\alpha\|_1$
 requires the α vector to be k -sparse, this formulation intuitively allows
 a linear rather than quadratic penalty to be placed on k of the residuals
 (with k tuned by choice of δ as expressed in the Huber loss formula-
 tion). These will correspond to those observations with the most le-
 verage (the most “outlying” points). We can then rewrite Eq. (18) as

$$\hat{y} = \arg \min (1/2) \|y - Z\gamma\|_2^2 + \lambda_G \gamma^T G' \gamma + \sum_{j=1}^{p+n} w_j |\gamma_j| \quad (19)$$

$$Z = [X \quad I_{n \times n}], \gamma = [\beta \quad \alpha], w_j = \begin{cases} \lambda_1 & j = 1, \dots, p \\ \delta & j = p + 1, \dots, p + n, \end{cases}$$

$$G' = \begin{bmatrix} G & 0_{1 \times n} \\ 0_{n \times 1} & 0_{n \times n} \end{bmatrix} \in S_+^{(p+n) \times (p+n)},$$

where $S_+^{m \times m}$ is the set of positive semidefinite $m \times m$ matrices. This is just
 a GraphNet problem in an augmented set of $p + n$ variables, and so can be
 solved using the fast coordinate-wise descent methods discussed in the
 section “Optimization and computational considerations” below. After

661 solving for augmented coefficients $\hat{\gamma}$ we can simply discard the last n coefficients to yield $\hat{\beta}$. A similar approach can be taken with the hinge-loss
 662 of a support vector machine classifier (as we show next), or more generally
 663 with any loss function decomposable into an infimal convolution
 664 of convex functions (see Appendix A). The Adaptive Robust GraphNet is
 665 easily obtained by letting

$$w_j = \begin{cases} \lambda_1^* \hat{w}_j & j = 1, \dots, p \\ \delta & j = p + 1, \dots, p + n \end{cases}$$

668 in Eq. (19) (see the section “Adaptive GraphNet regression” for more
 669 details on adaptive estimation).

670 *Huberized Support Vector Machine (SVM) GraphNet for classifications*

671 In the $p \gg n$ classification problem, maximum-margin classifiers
 672 such as the support vector machine (SVM) often perform exceedingly
 673 well in terms of classification accuracy, but do not yield readily interpretable
 674 coefficients. For this reason we also developed a sparse SVM
 675 with graph constraints, the Support Vector GraphNet (SVGNet), related
 676 to the “Hybrid Huberized SVM” of Wang et al. (2008a) as an alternative
 677 to the SPDA method. Using a “Huberized-hinge” loss function R_{HH}
 678 (see below) on the fit residuals, we have

$$\hat{\beta} = \kappa \arg \min R_{HH}(y^T X \beta; \delta) + \lambda_1 \|\beta\|_1 + \lambda_G \|\beta\|_G^2, \quad (20)$$

680 where $y \in \{-1, 1\}$, and letting $\hat{y} = X \hat{\beta}$ be the estimates of the target
 681 variable,

$$R_{HH}(y, \hat{y}; \delta) = n^{-1} \sum_{i=1}^n L_\delta(y_i, \hat{y}_i) \quad (21)$$

682 where $L_\delta(y_i, \hat{y}_i) = \begin{cases} (1 - y_i \hat{y}_i)^2 / 2\delta & \text{for } 1 - \delta < y_i \hat{y}_i \leq 1 \\ 1 - y_i \hat{y}_i - \delta / 2 & \text{for } y_i \hat{y}_i \leq 1 - \delta \\ 0 & \text{for } y_i \hat{y}_i > 1, \end{cases}$

683 which is the Huberized-hinge loss of Wang et al. (2008a). As with
 684 the Huber loss, there is an additional hyperparameter δ to be estimated
 685 or assumed. In this case, δ determines where the hinge-loss function
 686 switches from the quadratic to the linear regime (see Fig. 2a). This problem's
 687 loss function can also be written using infimal convolution to yield
 688 a more convenient quadratic objective term (see Appendix A). Finally,
 689 we discuss a heuristic alternative to adaptive methods for adjusting
 690 nonzero coefficient magnitudes to match the scale of the original data.
 691 This approach can be used with any of the above methods.

694 *Effective degrees of freedom for GraphNet estimators*

695 Following results for the Lasso Zou et al. (2007) and the Elastic Net
 696 der Kooij (2007), the effective degrees of freedom $\hat{d}f$ for the GraphNet
 697 regression are given by the trace of the “hat matrix” $H_{\lambda_G}(A)$ for the
 698 GraphNet estimator:

$$\hat{d}f = \text{tr}(H_{\lambda_G}(A)) = \text{tr}\left(X_A (X_A^T X_A + \lambda_G G)^{-1} X_A^T\right),$$

699 where X_A denotes the columns of X containing just the “active set”
 700 (those variables with nonzero coefficients corresponding to a particular
 701 choice of λ_1). This quantity is very useful in calculating standard
 702 model selection criteria such as the Akaike Information Criterion (AIC),
 703 Bayesian Information Criterion (BIC), Mallows’s C_p , Generalized Cross
 704 Validation (GCV), and others. Importantly, it can also be used for the various
 705 GraphNet methods, as each of these is solved as an equivalent GraphNet
 706 problem (for example, Eq. (19)) for the Adaptive Robust GraphNet.
 707

708 *Rescaling coefficients to account for “double shrinking”*

709 The Elastic Net was originally formulated by Zou and Hastie (2005) in
 710 both “naive” and rescaled forms. The authors noted that a combination of

ℓ_1 and ℓ_2 penalties can “double shrink” the coefficients. To correct this
 711 they proposed rescaling the “naive” solution by a factor of $\kappa = 1 + \lambda_2$
 712 (Zou and Hastie, 2005). Heuristically, the aim is to retain the desirable
 713 variable selection properties of the Elastic Net while rescaling the
 714 coefficients to be closer to the original scale. However, as this result is derived
 715 for an orthogonal design, it is not clear that $\kappa = 1 + \lambda_2$ is the
 716 correct multiplicative factor if the data are collinear, and this can complicate
 717 the problem of choosing a final set of coefficients. Following the arguments
 718 of Zou and Hastie (2005), for GraphNet regression we might rescale each
 719 coefficient by $\kappa_j = k(\hat{\Sigma}_{jj} + \lambda_G G_{jj})$ where k is the number of iterations
 720 used in the coordinate-wise descent optimization (and thus the number of
 721 times shrinkage related to G is applied, see Eq. (29) and derivations in
 722 Appendix A) and where $\hat{\Sigma} = X^T X$. In the case of an orthogonal design
 723 and $G = I$ we would have $\hat{\Sigma} = 1$ and thus $\kappa_j = 1 + \lambda_G$ —reducing to the
 724 Elastic Net rescaling employed in Zou and Hastie (2005).
 725

A simpler alternative is to fit the Elastic Net, generating a fitted response
 727 \hat{y} , and then to regress y on \hat{y} . In particular, solving the simple linear
 728 regression problem
 729

$$y = \kappa \hat{y} = \kappa X \hat{\beta}, \kappa \in \mathbb{R}$$

yields an estimate $\hat{\kappa}$ that can be used to rescale the coefficients
 730 obtained from fitting the Elastic Net (Daniela Witten and Robert Tibshirani,
 731 personal communication). The intuitive motivation for this heuristic is that
 732 it will produce a $\hat{\kappa}$ that puts $\hat{\beta}$ and \hat{y} on a reasonable scale for fitting y .
 733

Besides its simplicity, the principal advantage of this approach is that
 734 it requires no analytical knowledge about the amount of shrinkage that occurs
 735 as λ_G is increased. This is particularly appealing because the same strategy
 736 of regressing \hat{y} on y can be used with more general problems with more
 737 complicated forms, such as the Adaptive Robust GraphNet, where the
 738 additional shrinkage caused by the graph penalty can be corrected in this way.
 739

Finally, we note that over-shrinking is not necessarily bad for classification
 740 accuracy. Indeed it may improve accuracy due to the rather complicated
 741 relationship between bias and variance in the classification (for an excellent
 742 discussion in the context of 0–1 loss see Friedman (1997)). The focus on
 743 recovering good estimates of coefficient magnitude in this section is thus most
 744 relevant to regression and to situations in which correct estimates of
 745 coefficient magnitude are important.
 746

750 *Interpreting GraphNet regression and classification*

751 *Interpreting GraphNet parameters: dual variables as prices*

752 The GraphNet problem expressed in Eq. (5) derives from a constrained
 753 maximum likelihood problem, in which we want to maximize the likelihood
 754 of the parameters given the data, subject to some hard constraints on the
 755 solution—specifically, that they are sparse and structured (in the sense that
 756 their ℓ_1 and graph-weighted ℓ_2 norms are less than or equal to some
 757 constraint size). For concave likelihoods (as in generalized linear models
 758 and the cases considered above), this is a constrained convex optimization
 759 problem
 760

$$\text{maximize}_{\beta} \quad \text{loglik}(\beta|X, y) \quad (22)$$

$$\text{subject to} \quad \|\beta\|_1 \leq c_1 \quad (23)$$

$$\|\beta\|_G^2 \leq c_G, \quad (24)$$

where $c_1 \in \mathbb{R}_+$ and $c_G \in \mathbb{R}_+$ set hard bounds on the size of the coefficients
 762 in the ℓ_1 and ℓ_G norms, respectively. A standard approach for solving
 763 such problems is to relax the hard constraints to linear penalties
 764

769 (Boyd and Vandenberghe, 2004) and consider just those terms contain- 822
 770 ing β , giving the “Lagrangian” form of the GraphNet problem 823

$$\hat{\beta} = \arg \min_{\beta} -\log \text{lik}(\beta|X, y) + \lambda_1 \|\beta\|_1 + \lambda_G \|\beta\|_G^2, \lambda_1, \lambda_G \in \mathbb{R}_+, \quad (25)$$

772 which contains a negative likelihood term that measures misfit to the 824
 773 data as well as the two penalties characteristic of GraphNet estimators. 825

774 In this Lagrangian formulation, the dual variables λ_1 and λ_G repre- 826
 775 sent (linear) costs in response to a violation of the constraints. Since 827
 776 we solve problem (25), c_1 and c_G are effectively zero, and we are pe- 828
 777 nalized for any deviation of the coefficients from zero. This leads to 829
 778 one interpretation of λ_1 and λ_G : they are prices that we are willing 830
 779 to pay to improve the likelihood at the expense of a less sparse or 831
 780 less structured solution, respectively. For this reason, examining fit 832
 781 sensitivity to different values of λ_1 and λ_G tells us about underlying 833
 782 structure in the data. For example, if the task-related neural activity 834
 783 was very sparse and highly localized in a few uncorrelated voxels, 835
 784 then we should be willing to pay more for sparsity and less for 836
 785 smoothness (i.e., large λ_1 , small λ_G). In contrast, if large smooth and 837
 786 correlated regions underlie the task, then tolerating a large λ_G could 838
 787 substantially improve the fit. To explore such possibilities, we can 839
 788 plot test rates from cross validations at different combinations of pa- 840
 789 rameters. Fig. 6 shows plots of median test classification rates as a 841
 790 function of λ_1 and λ_G over the parameter grid on which the various 842
 791 GraphNet classifiers were fit. We see that there are regions in the 843
 792 (λ_1, λ_G) parameter space that clearly result in better median classifica- 844
 793 tion test rates, corresponding to fits with particular levels of smooth- 845
 794 ness and sparsity. 846

795 *Interpreting GraphNet coefficients*

796 Problem (25) can also be arrived at from a Bayesian perspective as 847
 797 a maximum a posteriori (MAP) estimator. In this case, the form of the 848
 798 penalty $P(\beta)$ is related to one’s prior beliefs about the structure of the 849
 799 coefficients. For example, under the well-known equivalence of pe- 850
 800 nalized regression techniques and posterior modes, the Elastic Net 851
 801 penalty corresponds to the prior

$$p_{\lambda_1, \lambda_2}(\beta) \propto \exp\left\{-\left(\lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2\right)\right\}$$

802 (Zou and Hastie, 2005). The GraphNet penalty thus corresponds to 852
 803 the prior distribution 853

$$p_{\lambda_1, \lambda_G}(\beta) \propto \exp\left\{-\left(\lambda_1 \|\beta\|_1 + \lambda_G \beta^T G \beta\right)\right\} \quad (26)$$

$$\propto \prod_{i=1}^p \exp\left\{-\lambda_1 |\beta_j|\right\} \prod_{i=1}^p \exp\left\{-\lambda_G \sum_{i \sim j} \beta_i G_{ij} \beta_j\right\},$$

806 where $i \sim j$ denotes that node i in the graph G is adjacent to node j . 854
 807 Therefore, the GraphNet problems are also equivalent to a MAP esti- 855
 808 mator of the coefficients with a prior consisting of a convex combina- 856
 809 tion of a global Laplacian (double-exponential) and a local Markov 857
 810 Random Field (MRF) prior. In other words, GraphNet methods explic- 858
 811 itly take into account prior beliefs about coefficients being globally 859
 812 sparse but locally structured. 860

813 *Optimization and computational considerations*

814 *Coordinate-wise descent and active set methods*

815 Fitting regression methods to whole-brain fMRI data requires effi- 861
 816 cient computational methods, particularly when they must be cross- 862
 817 validated over a grid of possible parameter values. For instance, in 863
 818 the shopping example described in greater detail below (see the sec- 864
 819 tion “Application: predicting buying behavior using fMRI”), 26,630 865
 820 input features (voxels) at each of 7 time points are used to classify fu- 866
 821 ture choices to purchase a product or not. Fitting the Adaptive Robust 867

GraphNet using leave-one-subject-out (LOSO) cross-validation (i.e., 822
 25 fits) for each realization of possible parameter values over this 823
 90 × 5 × 6 × 10 × 3 grid of possible parameters $\{\lambda_1, G, \lambda_G, \delta, \lambda_1^*\}$ requires 824
 2,025,000 model fits on 1882 observations of 186,410 input features. 825

To efficiently fit GraphNet methods with millions of parameter 826
 combinations over hundreds of thousands of input features, we for- 827
 mulated the minimization problem (i.e., Eqs. (5), (15), and (20)) as 828
 a coordinate-wise optimization procedure (Tseng, 1988, 2001) using 829
 active set methods (Friedman et al., 2010). This approach fit one coef- 830
 ficient value at a time (“coordinate-wise” descent), holding the rest 831
 constant, and kept an “active set” of nonzero coefficients. Fitting was 832
 initiated with a large value of λ_1 (corresponding to all coefficients 833
 being zero), and then slowly decreased λ_1 to allow more and more co- 834
 efficients into the model fit. This procedure thus considered an “active 835
 set” of the model coefficients at each coordinate-wise update, rather 836
 than all 186,410 inputs. Occasional sweeps though all the coefficients 837
 were made to search for new variables to include, as in Friedman et al. 838
 (2010). Model fitting terminated before λ_1 reached zero, since fitting a 839
 fully dense set of coefficients is computationally expensive and known 840
 to produce poor estimates (Friedman et al., 2010; Hastie et al., 2009). 841
 Various heuristics and model selection criteria may be used for choos- 842
 ing a stopping point, for example, stopping once the AIC or BIC for the 843
 model stops decreasing and starts increasing. AIC is known to be 844
 over-inclusive in model selection, and is therefore a more conserva- 845
 tive stopping point. 846

Coordinate-wise descent is guaranteed to converge for GraphNet 847
 methods because they are all of the form 848

$$\arg \min_{\beta} f(\beta_1, \dots, \beta_p) = \arg \min_{\beta} g(\beta_1, \dots, \beta_p) + \sum_{j=1}^p h(\beta_j), \quad (27)$$

where $g(\beta_1, \dots, \beta_p)$ is a convex, differentiable function (e.g., squared- 849
 error and Huber loss plus the quadratic penalty $\|\beta\|_G^2$), and where 850
 each $h(\beta_j)$ is a convex (but not necessarily differentiable) function 851
 (e.g., the ℓ_1 penalty). If the convex, non-differentiable part of the 852
 penalty function is separable in coordinates β_j (as is true of $\|\beta\|_1 = 853$
 $\sum_{j=1}^p |\beta_j|$), then coordinate descent converges to a global solution of 854
 the minimization problem (Tseng, 2001). In the case of Huber loss or 855
 Huberized-hinge loss, the two-part loss function can be written as a 856
 single quadratic loss function using infimal convolution as described 857
 in the section “Huberized Support Vector Machine (SVM) GraphNet 858
 for classifications”. For instance, consider the coordinate-wise updates 859
 for the standard GraphNet problem given in Eq. (5). Letting $\tilde{y} = \tilde{X}\tilde{\beta} + 860$
 $X_{\cdot j}\beta_j$ (where $\tilde{X} = X_{\cdot \neq j}$ is the matrix X with the j th column removed, 861
 and $\tilde{\beta} = \beta_{\neq j}$ the coefficient vector with the j th coefficient removed), 862
 the subdifferential of the risk with respect to only the j th coefficient 863
 (holding the others fixed) is 864
 865

$$\partial_{\beta_j} R_p = -X_{\cdot j}^T y + X_{\cdot j}^T \tilde{X} \tilde{\beta} + X_{\cdot j}^T X_{\cdot j} \beta_j + (\lambda_2/2) \tilde{\beta}^T (G_{\neq j \cdot})_{\cdot j} + \lambda_2 G_{jj} \beta_j$$

$$+ (\lambda_1/2) \Gamma(\beta_j), \quad (28)$$

where the set-valued function $\Gamma(\beta_j) = -1$ if $\beta_j < 0$, $\Gamma(\beta_j) = 1$ if $\beta_j > 0$ and 866
 $\Gamma(\beta_j) \in [-1, 1]$ if $\beta_j = 0$. If we let $\Gamma(\beta_j) = \text{sign}(\beta_j)$ in Eq. (28) (which is al- 867
 ways a particular subgradient in the subdifferential of the risk), then 868
 the coordinate update iteration for the j th coefficient estimate is 869
 870

$$\hat{\beta}_j \leftarrow \frac{S(X_{\cdot j}^T (y - \tilde{X} \tilde{\beta}) - (\lambda_2/2) \tilde{\beta}^T (G_{\neq j \cdot})_{\cdot j}, \lambda_1/2)}{X_{\cdot j}^T X_{\cdot j} + \lambda_2 G_{jj}}, \quad (29)$$

where 871

$$S(x, \gamma) = \text{sign}(x)(|x| - \gamma)_+ \quad (30)$$

is the soft-thresholding function (Donoho, 1995; Friedman et al., 872
 2007a). Note that if graph $G = I$, and the columns of X are standardized 873

to have unit norm, then the coordinate-wise Elastic Net update is recovered (der Kooij, 2007; Friedman et al., 2007b).

Computational complexity

A closer look at Eq. (29) reveals that if the variables are standardized (such that $X_j^T X_j = 1$) then the $(c + 1)$ st coefficient update for the j th coordinate can be rewritten

$$\hat{\beta}_j^{(c+1)} \leftarrow -S \left(\sum_{i=1}^N x_{ij} r_i^{(c)} + \hat{\beta}_j^{(c)} - (\lambda_2/2) \sum_{k \neq j} \beta_k G_{kj}, \lambda_1/2 \right) / (1 + \lambda_2 G_{jj}), \quad (31)$$

where $r = y - \hat{y}$ is the vector of residuals. Letting m be the number of off-diagonal nonzero entries in G and initializing with $\hat{\beta}_j^{(0)} = 0$ for all j and $r^{(0)} = y$, the first sweep through all p coefficients will take $O(pn) + O(m)$ operations. Once a_1 variables are included in the active set, q iterations are performed according to Eq. (31) until the new estimates converge, at which point λ_1 is decreased incrementally and another $O(pn)$ sweep is made through the coefficients to find the next active set with a_2 variables (using the previous estimate as a warm start to keep q small). This procedure is repeated for l values of λ_1 , until the fit stops improving or a pre-specified coefficient density is reached. Let $a = \sum_{i=1}^l a_i$ denote the total number of coefficient updates over all l fits. The total computational complexity is then $O(lpn) + O(lm) + O(aq)$. Thus if G is relatively sparse (so m is small) and if it requires few iterations for coefficients in an active set to converge (q small)—which is true if the unpenalized loss function is quadratic—then the computational complexity is dominated by the $O(lpn)$ term representing the sweep through the coefficients necessary to find the next active set for each new value of λ_1 . We note that this suggests that including a screening procedure such as the STRONG rules (Tibshirani et al., 2012) could further speed up fitting in this context. Either making G dense or decreasing λ_1 until a becomes large can cause the other complexity terms to play a significant role and slow the speed of the algorithm. For example, if G is dense, then $m = p^2 - p$ and the $O(lm)$ term will dominate.

Cross validation, classification accuracy, and parameter tuning

For training and test data, trials for each subject were resampled within-subject to consist of 80 trials with exactly 40 purchases. If the subject originally had more than 40 purchases, sampling without replacement was used to select 40. If the subject originally had fewer than 40 purchases, sampling with replacement was used to select 40. Similar sampling was used to select exactly 40 trials without purchases. This resampling scheme ensured that the trials for each subject were balanced between purchasing and not purchasing. Further, because our cross-validation schemes defined folds on the subject level, this ensured that every training and test set in the cross-validation were also balanced.

For the cross-validation, the range for these grid values was chosen based off of a few preliminary fits. This grid is very large, and with the refitting involved in cross-validation, resulted in millions of fits. The smoothness of the rates as a function of the parameters (see Fig. 6) suggests that smaller grids are likely better suited to most applications, and we anticipate that more efficient adaptive approaches to parameter search—such as focused grid search methods (Jimenez et al., 2009) or sampling methods inspired by Bayesian approaches to similar problems—will ultimately prove superior. We leave these refinements to future work. The grid values used here are given in Appendix A.

In order to choose a final set of coefficient estimates from multiple fits across cross-validation folds, we took the element-wise median of the coefficient vectors across the folds. Thus a feature corresponding to a particular voxel at a particular TR would have to appear (be non-zero) in more than half of the 25 cross-validation folds in order to be included in the final coefficient estimate used in the out-of-sample (OOS) analysis. There are several justifications for taking the median

across folds: (1) the median preserves sparsity, (2) the median is the appropriate maximum likelihood estimator for the double-exponential (Laplacian) distribution that corresponds to the ℓ_1 sparsity prior on the coefficients (see discussion in Grosenick et al., 2008), (3) such a procedure is closely related to the Median Probability Model, which is the model consisting of those variables that have posterior probability ≥ 0.5 of being in a model, and which has been shown to have optimal predictive performance for linear models (Barbieri and Berger, 2004), and (4) it is similar to other recently-developed model selection procedures for sparse models such as Stability Selection Meinshausen and Bühlmann (2010) that use the number of times a variable appears across multiple sparse fits to resampled data in order to significantly improve model selection. Further, we have found this approach to be quite effective in practice (see the out-of-sample results that follow). Note that such inclusion of a variable only if it appears in more than half of the 25 cross-validation folds is a natural means of imposing some “reliability” or “stability” on the coefficients.

Application: predicting buying behavior using fMRI

Subjects and SHOP task

Data from 25 healthy right-handed subjects were analyzed (Knutson et al., 2007). Along with the typical magnetic resonance exclusions (e.g., metal in the body), subjects were screened for psychotropic drugs, cardiac drugs, ibuprofen, substance abuse in the past month, and history of psychiatric disorders (DSM IV Axis I) prior to collecting informed consent. Subjects were paid \$20.00 per hour for participating and also received \$40.00 in cash to spend on products. Of 40 total subjects, 6 subjects who purchased fewer than four items per session (i.e., <10%) were excluded due to insufficient data to fit, 8 subjects who moved excessive amounts (i.e., > 2 mm between whole brain acquisitions) were excluded, and one subject's original fMRI data could not be recovered and so was omitted, yielding the final total of 25 subjects included in the analysis.

While being scanned, subjects participated in a “Save Holdings Or Purchase” (SHOP) task (Fig. 3). During each task trial, subjects saw a labeled product (product period; 4 s), saw the product's price (price period; 4 s), and then chose either to purchase the product or not (by selecting either “yes” or “no” presented randomly on the right or left side of the screen; choice period; 4 s), before fixating on a crosshair (2 s) prior to the onset of the next trial (see Fig. 3).

Each of 80 trials featured a different product. Products were pre-selected to have above-median attractiveness, as rated by a similar sample in a pilot study. While products ranged in retail price from \$8.00 to \$80.00, the associated prices that subjects saw in the scanner were discounted down to 25% of retail value to encourage purchasing. Therefore the cost of each product during the experiment ranged from \$2.00 to \$20.00. Consistent with pilot findings, this led subjects to purchase 30% of the products on average, generating sufficient instances of purchasing to fit.

To ensure subjects' engagement in the task, two trials were randomly selected after scanning to count “for real”. If subjects had chosen to purchase the product presented during the randomly selected trial, they paid the price that they had seen in the scanner from their \$40.00 endowment and were shipped the product within two weeks. If not, subjects kept their \$40.00 endowment. Based on these randomly drawn trials, seven of twenty-five subjects (28%) were actually shipped products.

Subjects were instructed in the task and tested for comprehension prior to entering the scanner. During scanning, subjects chose from 40 items twice and then chose from a second set of 40 items twice (80 items total), with each set presented in the same pseudo-random order (item sets were counterbalanced across subjects). We consider only data from the first time each item was presented here (see Grosenick et al. (2008) for a comparison between first and second



Fig. 3. Save Holdings, or Purchase (SHOP) task trial structure. Images represent what the subject saw, bars represent 2 s TRs (T1–T7). Subjects saw a labeled product (product period; 4 s, 2 TRs), saw the product's price (price period; 4 s, 2 TRs), and then chose either to purchase the product or not (by selecting either "yes" or "no" presented randomly on the right or left side of the screen; choice period; 4 s, 2 TRs), before fixating on a crosshair (2 s, 1 TR) prior to the onset of the next trial.

presentations). After scanning, subjects rated each product in terms of how much they would like to own it and what percentage of the retail price they would be willing to pay for it. Then, two trials were randomly drawn to count "for real", and subjects received the outcome of each of the drawn trials.

A second validation sample included 17 healthy right-handed subjects (Karmarkar et al., submitted for publication). These subjects passed the same screening, inclusion, and exclusion criteria. Of an original sample of 24, 6 subjects purchased fewer than four items per session, and one showed excessive motion. These subjects were excluded from analyses, as before. Subjects also received the same payment and underwent the same scanning and experimental procedures. Importantly, however, subjects were different individuals who were exposed to different products, and were scanned more than three years after the original study.

Image acquisition

Functional images were acquired with a 1.5 T General Electric MRI scanner using a standard birdcage quadrature head coil. Twenty-four 4-mm-thick slices (in-plane resolution 3.75×3.75 mm, no gap) extended axially from the midpons to the top of the skull, providing whole-brain coverage and adequate spatial resolution of subcortical regions of interest (e.g., midbrain, NAcc, OFC). Whole-brain functional scans were acquired with a T2*-sensitive spiral in-/out- pulse sequence (TR = 2 s, TE = 40 ms, flip = 90), which minimizes signal dropout at the base of the brain (Glover and Law, 2001). High-resolution structural scans were also acquired to facilitate localization and coregistration of functional data, using a T1-weighted spoiled grass sequence (TR = 100 ms, TE = 7 ms, flip = 90).

Preprocessing

After reconstruction, preprocessing was conducted using Analysis of Functional Neural Images (AFNI) software (Cox, 1996). For all functional images, voxel time-series were sinc interpolated to correct for non-simultaneous slice acquisition within each volume, concatenated across runs, corrected for motion, and normalized to percent signal change with respect to the voxel mean for the entire task. For further preprocessing details see (Grosenick et al., 2008). Given that spatial blur would artificially increase correlations between variables for the voxel-wise analysis, we used data with no spatial blur and a temporal high pass filter for all analyses. Note that in general, smoothing before running analyses will compound the problems with correlation mentioned above, resulting in "rougher" (high-frequency) coefficients overall (see discussion in (Hastie et al., 1995)).

Spatiotemporal data were arranged as in previous spatiotemporal analyses (Mourão-Miranda et al., 2007). Specifically, data was arranged as an $n \times p$ data matrix X with n corresponding to the number of trial observations on the p input variables, each of which was a

particular voxel at a particular time point. This yielded 26,630 voxels taken at 7 time points (each taken every 2 s), yielding a total of $p = 186,410$ input features per trial. Altogether, the data used for training and test from (Knutson et al., 2007) included $n = 1882$ trials across the 25 subjects. The validation sample from (Karmarkar et al., submitted for publication) included $n = 322$ trials across the 17 subjects. In the first case (training and testing on the Knutson et al. (2007) data), the number of 'buy' trials were upsampled to match the number of 'not buy' trials in order to efficiently use the data when fitting the models. In the out-of-sample (OOS) validation on the (Karmarkar et al., submitted for publication) data, however, the number of 'not buy' trials were downsampled to match the smaller number of 'buy' trials in order to be more conservative in estimating the out-of-sample accuracy (and related p -values).

Results

Classification rates

If neural substrates implicated in choice show invariance across individuals, a method that successfully identifies and uses these substrates to predict choice should generalize well across subjects. We compared the GraphNet classifier accuracies with accuracies obtained using linear SVM (where accuracy in this case is the ability to correctly predict a subject's choices to purchase a product or not). In particular we looked at generalization of fits to held-out "test" sets (consisting of subjects held out of a particular stage of the cross validation procedure, but still present in other cross-validation stages), and to out-of-sample (OOS) data (new data never used at any stage of the model fitting) consisting of different subjects from another study (Karmarkar et al., submitted for publication). Results and model parameters for the GraphNet classifiers and linear SVM across the 25 subjects from Knutson et al. (2007) ("Training", "Test") and 17 subjects from Karmarkar et al. (submitted for publication) ("OOS") are listed in Tables 1 and 2, as well as a summary of each method's properties. Models were fit using either leave-one-subject-out (LOSO, Table 1) or leave-5-subjects-out (L5SO, Table 2) cross-validation, and both training and test results are displayed to allow comparison of overfitting on the training data versus the held-out test data. As cross-validation is known to yield an overly optimistic estimate of the true classification error rate (Hastie et al., 2009), model fits to the initial data set ($n = 25$; Knutson et al. (2007)) were tested on out-of-sample (OOS) data ($n = 17$; Karmarkar et al. (submitted for publication)) collected more than three years later using different subjects shown different products. These out-of-sample results provide the most rigorous demonstration of fit generalization to new data, adjusting for any over fitting by the cross-validation procedure, and are the strongest evidence for invariance in the neural

Table 1
Median classification accuracy and parameters for SPDA and SVM classifiers fit with leave-5-subjects-out (L5SO) cross validation.

Method	Classification accuracy				Model type				
	Training	Test	OOS	p-Value [†]	Sparse	Tikhonov	Structured	Robust	Adaptive
					(λ_1)	(λ_2)	(λ_C)	(δ)	(λ_1^*)
Linear SVM ^a	97.9%	71.0%	65.8%	2.7×10^{-8}		3.8×10^{-6}		✓	
Lasso ^b	98.8%	68.5%	58.4%	0.003	33				
Elastic Net ^c	90.4%	72.5%	64.3%	3.3×10^{-7}	54	10000			
GraphNet ^d (GN)	86.9%	73.7%	64.6%	1.8×10^{-7}	68	1000	100		
Robust GN (RGN)	86.8%	74.5%	64.9%	1.8×10^{-7}	43	100	100	0.3	
RGN + temporal	96.5%	73.8%	63.0%	5.7×10^{-6}	42	1000	10	0.5	
Adaptive RGN	91.4%	73.8%	67.1%	8.6×10^{-10}	50	10000	100	0.4	0.01
ARGN + temporal	90.8%	73.5%	66.8%	1.8×10^{-9}	40	1000	100	0.3	0.01
Support Vector GN	85.3%	73.0%	62.4%	1.6×10^{-5}	120	1000	10	0.5	

OOS is short for “out-of-sample”. Chance level is 50%. [†]p-Value is calculated for the out-of-sample accuracy using an exact test for the probability of success in a Bernoulli experiment with $n = 322$ trials with success probability of 0.5. ^{††}This is the C parameter for the SVM. ^{†††}The linear SVM is robust as a result of its hinge loss function, which does not have a parameter δ associated with it.

^a (Cortes and Vapnik, 1995).
^b (Tibshirani, 1996).
^c (Zou and Hastie, 2005).
^d (Grosenick et al., 2009b).

1091 representation of choice across subjects. The p -values reported corre- 1114
 1092 spond to these out-of-sample accuracies on $n = 322$ trials across the 1115
 1093 17 new subjects. 1116

fits to held-out subjects. The marked non-normality of these distribu- 1114
 tions is interesting, and motivated us to report median rather than 1115
 mean accuracy over cross-validation folds. 1116

1094 *Generalization to held-out groups (L5SO cross-validation)*

Generalization to held-out individuals (LOSO cross-validation) 1117

1095 Best median training, median test, and out-of-sample (OOS) rates 1118
 1096 are described for GraphNet classifiers fit over the grid of parameters 1119
 1097 given in Eq. (33). The linear SVM parameters are also given in Eq. 1120
 1098 (33). Despite a more than 1000-fold increase in the number of input 1121
 1099 features relative to earlier volume of interest (VOI) analyses 1122
 1100 (Grosenick et al., 2008), whole-brain classifiers performed significant- 1123
 1101 ly better than previous VOI-based predictions fit to the same data 1124
 1102 (Grosenick et al., 2008; Knutson et al., 2007). Further, among these 1125
 1103 whole-brain classifiers, adaptive and robust methods performed best 1126
 1104 on out-of-sample data. SVGN performed similarly to the linear SVM 1127
 1105 (but unlike linear SVM, yields structured, sparse coefficients that aid 1128
 1106 interpretability). Further, Lasso and linear SVM tended to overfit the 1129
 1107 training data more than the SPDA-GraphNet classifiers, as evidenced 1130
 1108 by their higher training but lower test rates. Overall, the Adaptive Rob- 1131
 1109 ust GraphNet classifier showed the best out-of-sample classification 1132
 1110 rate, with accuracy on new data of 67.1% (for comparison, the linear 1133
 1111 SVM accuracy was 65.8%). Examining the distribution of test classifica- 1134
 1112 tion rates across the 25 folds (25 sets leaving 5 subjects out), Fig. 4b 1135
 1113 shows that the linear SVM appears to have less variance across test

In addition to the leave-5-subjects out (L5SO) cross-validation, we 1118
 also ran leave-one-subject-out (LOSO) cross validation (i.e., using 1119
 the data from 24 subjects to predict results for each remaining sub- 1120
 ject). Repeating this procedure for all subjects yielded one held-out 1121
 classification rate per subject, indicating how well the group fit gener- 1122
 alized to that subject. Repeating this for all subjects yielded one held- 1123
 out test rate per subject. This rate indicated how well the model fit 1124
 based on all but one subjects' data generalized to the held-out 1125
 subject—a measure of invariance across subjects as well as a quantity 1126
 that may be of interest in studies of individual differences. Fig. 4a 1127
 shows smoothed histograms of the LOSO classification rates for the 1128
 Robust GraphNet, Adaptive Robust GraphNet, SVGN, and linear SVM 1129
 classifiers. Overall, the GraphNet classifiers outperform the linear 1130
 SVM on LOSO cross-validation across subjects. When the LOSO fits 1131
 were used to classify choice out-of-sample, the Adaptive Robust 1132
 GraphNet classifier again yielded the best performance, now at al- 1133
 most 70% classification accuracy. LOSO cross-validation appears to re- 1134
 sult in better OOS generalization than L5SO cross-validation for this 1135

Table 2
Median classification accuracy and parameters for SPDA and SVM classifiers fit with leave-one-subject-out (LOSO) cross validation.

Method	Classification accuracy				Model type				
	Training	Test	OOS	p-Value [†]	Sparse	Tikhonov	Structured	Robust	Adaptive
					(λ_1)	(λ_2)	(λ_C)	(δ)	(λ_1^*)
Linear SVM ^a	91.6%	68.8%	65.2%	9.7×10^{-8}		7.6×10^{-6}		✓	
Lasso ^b	90.5%	68.8%	61.2%	7.1×10^{-5}	63				
Elastic Net ^c	90.8%	70.0%	63.0%	5.7×10^{-6}	61	1000			
GraphNet ^d (GN)	87.5%	71.3%	67.7%	4.1×10^{-10}	54	10000	1000		
Robust GN (RGN)	83.8%	72.5%	67.4%	4.1×10^{-10}	25	10	100	0.2	
RGN + temporal	83.8%	72.5%	67.1%	8.6×10^{-10}	55	100	1000	0.6	
Adaptive RGN	85.4%	72.5%	69.8%	1.7×10^{-12}	20	10	1000	0.2	0.01
ARGN + temporal	88.3%	73.8%	68.9%	2.0×10^{-11}	30	1000	100	0.2	0.01
Support Vector GN	89.5%	73.8%	65.2%	9.7×10^{-8}	84	100	100	0.5	

OOS is short for “out-of-sample”. Chance level is 50%. [†]p-Value is calculated for the out-of-sample accuracy using an exact test for the probability of success in a Bernoulli experiment with $n = 322$ trials with chance level at 50%. ^{††}This is the C parameter for the SVM. ^{†††}The linear SVM is robust as a result of its hinge loss function, which does not have a parameter δ associated with it.

^a (Cortes and Vapnik, 1995).
^b (Tibshirani, 1996).
^c (Zou and Hastie, 2005).
^d (Grosenick et al., 2009b).

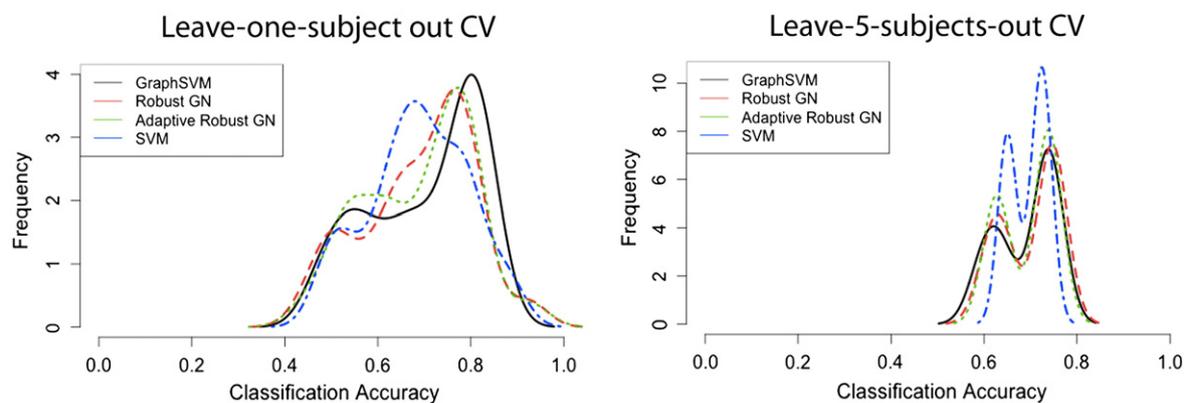


Fig. 4. (Left) Smoothed histogram densities of leave-one-subject out (LOSO) accuracy rates on test data. Models were fit to all subjects except one, and then tested on the held-out subject. This was done for all subjects and smoothed histograms of these rates were calculated for the best fitting models. (Right) The same procedure was repeated, but leaving 5 subjects out at a time for a total of 25 cross-validation folds. Both plots show some bi-modality suggestive of different underlying groups.

1136 data. More important than the improvement in classification perfor-
1137 mance, however, is the greater interpretability of these methods.

1138 Visualization and interpretation of coefficients and parameters

1139 Interpreting GraphNet coefficients

1140 While GraphNet classifiers and linear SVM both classified purchase
1141 choices successfully, the GraphNet-based classifiers produced more
1142 interpretable results. Consistent with previous VOI-based analyses, the
1143 GraphNet, Robust GraphNet classifier (Fig. 5), and Adaptive Robust
1144 GraphNet (Fig. 5) classifiers all identified similar regions to those chosen
1145 as VOIs (Knutson et al., 2007), with coefficients present at the time points
1146 corresponding to peak discrimination in the VOI time-series (Knutson et
1147 al., 2007) and VOI classification (Grosenick et al., 2008). In particular, nu-
1148 cleus accumbens (NAcc) activation began to positively predict purchase
1149 choices at the time of product presentation, and this prediction persisted
1150 throughout subsequent price presentation. Medial prefrontal cortex
1151 (MPFC) and midbrain activation, on the other hand, began to positively
1152 predict purchase choices at the onset of price presentation (but not dur-
1153 ing previous product presentation). Additionally—and not included in
1154 any previous findings—posterior cingulate activation also began to ro-
1155 bustly and positively predict purchase choices during price presentation.
1156 Reassuringly, no regions' activation predicted purchase choices during
1157 fixation presentation. Interestingly, the best fits chose far more voxels
1158 that positively predicted than negatively predicted purchasing.

1159 Together, these findings demonstrate that sparse, structured,
1160 whole-brain methods like GraphNet can facilitate the discovery of
1161 new behaviorally-relevant spatiotemporal neural activity patterns
1162 that existing VOI-based methods miss, particularly when made robust
1163 and adaptive. For example, given the temporal as well as spatial resolu-
1164 tion of the present design and data, it was possible to extend inter-
1165 pretation of the model fit not only to where brain activity predicted
1166 purchasing choices, but also to when and in what order, and to new re-
1167 gions not chosen as VOIs in previous work emerged and improved the
1168 overall classification. Thus, an investigator who knows when different
1169 events occurred (and accounts for the lag and variation of the peak he-
1170 modynamic response) can infer that different design components pro-
1171 moted eventual purchasing choices by altering activity in specific
1172 regions. The ability of coefficient vectors estimated from the Knutson
1173 et al. (2007) to accurately predict choices of new subjects run on the
1174 SHOP task years later and shown different products speaks both to
1175 the stability of the neural activity related to the task across subjects
1176 and products, and to the quality of the model.

1177 Interpreting GraphNet parameters

1178 Fig. 6 shows plots of median L5SO cross-validation rates over the
1179 values $\{\lambda_1, \lambda_C, G\}$ (28) on which we fit the four GraphNet classifiers

(other parameters are set to the values shown in Table 1; plots for
LOSO rates are similar). In all cases, there is a region in the interior of
the explored parameter space $\{\lambda_1, \lambda_C, G\}$ in which the models empirically
perform best. In all cases this region involves both smoothing and some
level of sparsity, and the classifiers built with Lasso (L) and Elastic Net
(EN)—shown as separate bars for the GraphNet (GN) fits—under-
perform relative to the sparse and smooth GraphNet classifiers on this
data. Comparison of the rates in Fig. 6 suggests that a certain amount
of coefficient smoothness and inclusion of correlated variables in the
final fit is important for this data set, and that using a robust loss func-
tion tightens the region of optimal parameter performance.

1191 Discussion and conclusions

1192 Interpretable models for whole-brain spatiotemporal fMRI

1193 We sought to design and develop a novel classification method for
1194 fMRI data that could fulfill several aims. First, the method should de-
1195 liver interpretable results for whole-brain data over multiple time-
1196 points in the native data space. Second, the method should yield
1197 classification accuracy (or goodness-of-fit) competitive with current
1198 state-of-the-art multivariate methods. Third, the method should
1199 choose relevant features in a principled and asymptotically consistent
1200 way (i.e., it should include relevant features while excluding nuisance
1201 parameters). Fourth, the method should accommodate flexible con-
1202 straints on model coefficients related to prior information (e.g., local
1203 smoothness, connectivity). Fifth, the method should remain robust
1204 to outliers in the data. Sixth, the method should generate coefficients
1205 with relatively unbiased magnitudes (despite employing shrinkage
1206 methods to yield sparsity). And seventh (and finally), the method
1207 should have the capacity to detect a range of possible signals, from
1208 smooth and localized to sparse and distributed.

1209 The GraphNet-based methods presented here make a first step to-
1210 ward meeting these desirable (and often competing) aims. In particu-
1211 lar, the Adaptive Robust GraphNet allows automatic variable selection
1212 (Zou and Zhang, 2009), incorporation of prior information in the form
1213 of a graph penalty, and yields minimally biased and asymptotically
1214 consistent coefficient estimates as a result of adaptive reweighting.
1215 Robust GraphNet methods can be applied to either regression or clas-
1216 sification settings (using Optimal Scoring), and generate classification
1217 rates that compete favorably with state-of-the-art multivariate classi-
1218 fiers. The tuning parameters (λ_1, λ_C) and the graph G allow for a diversity
1219 of sparse and smooth data, and the relationship of model fits to these pa-
1220 rameters provides information about the structure of the detected signal.

1221 Choice in the context of purchasing admittedly represents only one
1222 application, and future validation on additional data sets is necessary.
1223 However, in this context the GraphNet classifiers generalize well to

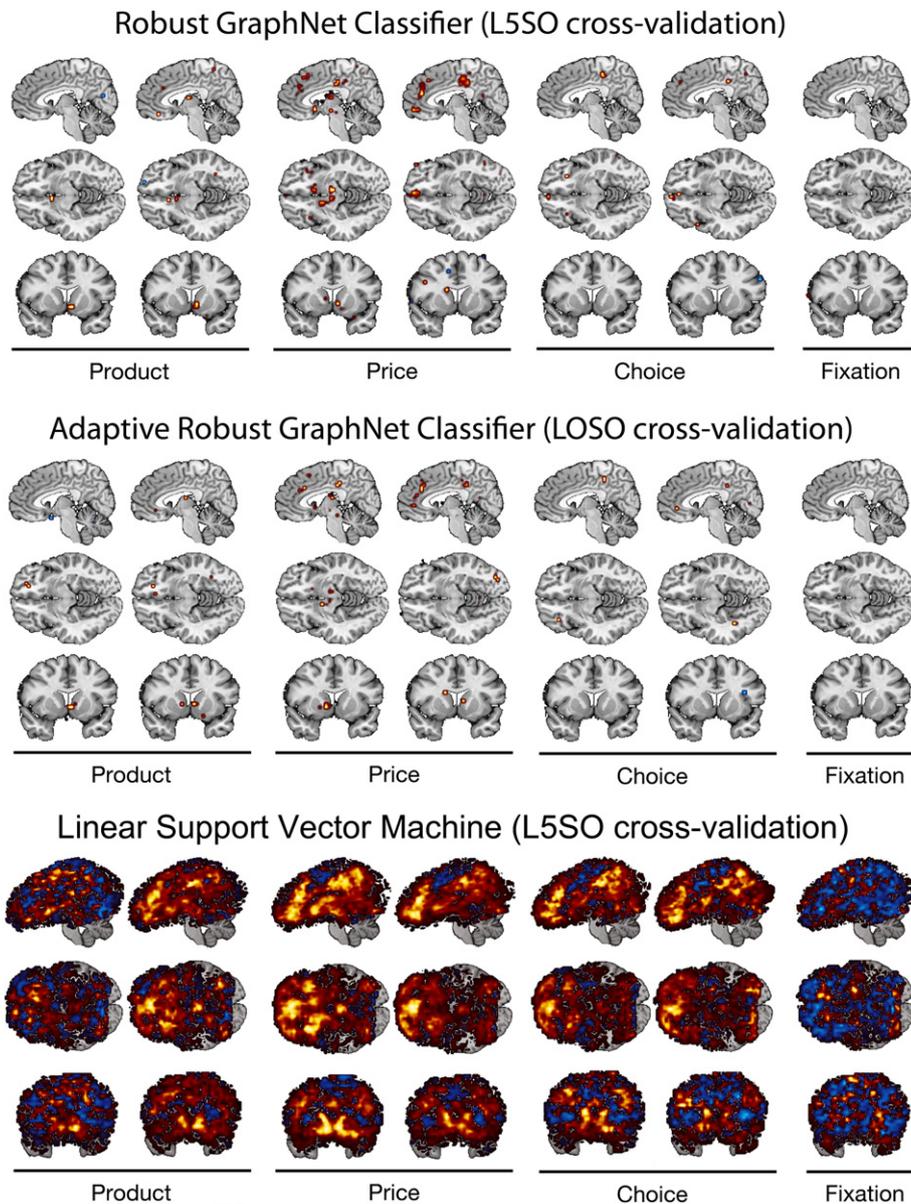


Fig. 5. Whole-brain classification results from the SHOP task (see Fig. 3 for task structure). (Top) Median coefficient maps from the best robust GraphNet classifier (median test accuracy of 74.5% over cross-validation folds and out-of-sample accuracy of 64.9%) fit using Leave-5-subjects-out (L5SO) cross-validation are shown at two time points for product, price, and choice periods, as well as the fixation period. Warm colored coefficients denote areas that predict purchasing a product, while cool-colored areas those that predict not purchasing. The areas chosen by the robust GraphNet classifier highlight regions suggested by previous studies including the bilateral nucleus accumbens (NAcc) and the mesial prefrontal cortex (MPFC) (Grosenick et al., 2008; Knutson et al., 2007), but also implicate new regions including the anterior cingulate and posterior cingulate cortices. (Middle) Similar plots for the best adaptive robust GraphNet classifier (median test accuracy of 72.5% over cross-validation folds; out-of-sample accuracy of 69.8%) fit using leave-one-subject (LOSO) cross-validation. Although the solution is sparser, the regions chosen remain the same. (Bottom) Coefficients for the best linear SVM (median test accuracy of 71% over cross-validation folds; out-of-sample accuracy of 65.8%) fit using Leave-5-subjects-out (L5SO) cross-validation for comparison.

1224 independent experiments involving purchasing (i.e. when fit to new
 1225 data collected years after the experiments originally used to train
 1226 the models, with different subjects and different products). Adaptive
 1227 Robust GraphNet methods showed the best out-of-sample general-
 1228 ization, and generated parsimonious, interpretable models. It is
 1229 worth noting that the models that did best in the in-sample cross-
 1230 validation test folds were not best in out-of-sample data. This suggests
 1231 that the overfitting, or “optimism”, known to exist in cross-validation
 1232 (Hastie et al., 2009) can effect models differently, and that a true out-
 1233 of-sample prediction is necessary to accurately assess which models
 1234 generalize best.

1235 In summary, we have developed a family of robust, adaptive, and inter-
 1236 pretable methods that can be fit efficiently to large data sets over
 1237 large parameter grids. This method will allow investigators to search
 1238 in a data-driven fashion across the whole brain and multiple time

1239 points, obviating the need for volume-of-interest based approaches in
 1240 fMRI classification and regression, and providing an effective alternative
 1241 to mass-univariate approaches for whole-brain analysis.

1242 Application to SHOP task data

1243 In the context of predicting human behavior from brain data, the
 1244 current whole brain methods offer clear advantages over previous vol-
 1245 ume of interest based methods. In terms of classification accuracy,
 1246 previous work on the Knutson et al. (2007) data has resulted in cross-
 1247 validated test rates of 60% (with a leave-one-out cross validation
 1248 using logistic regression on VOI-averaged data; see Knutson et al.
 1249 (2007) for details), and 67% (with a 5×2 cross validation using
 1250 SPDA-Elastic Net on VOI voxel data; see Grosenick et al. (2008) for de-
 1251 tails). Here, using the same preprocessing and data as in these previous

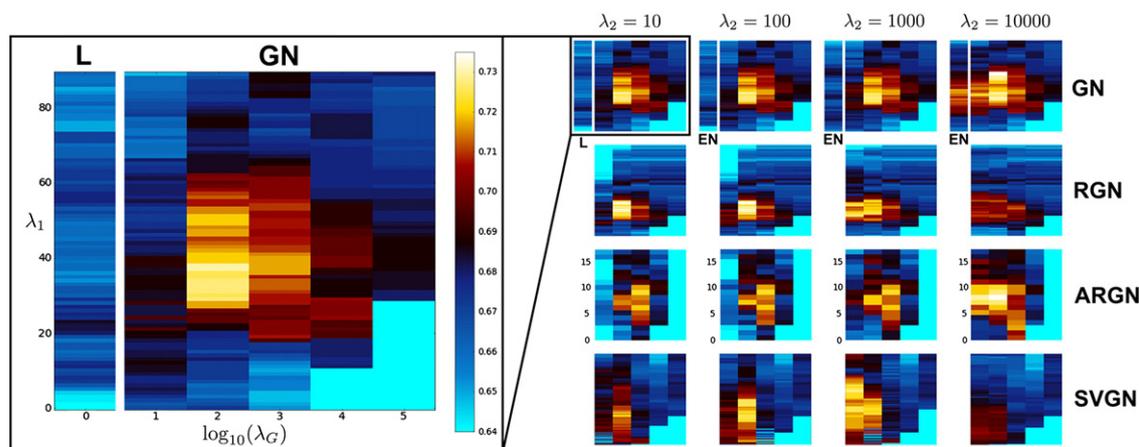


Fig. 6. Examples of classification accuracy (test) plotted as a function of penalty parameters. The blown up image on the left shows an image of the median test accuracy rates for the GraphNet SPDA classifier (GN) as functions of hyperparameters λ_1 and λ_G (with $\lambda_2 = 0$). Warm colors indicate median classification rates above 70% (for LSSO cross-validation) and cool colors median accuracy below 70% (see color bar for scale). The separate column (L) indicates the standard Lasso solution at $\lambda_G = 0$. There is a clear maxima at $\lambda_1 = 40$, $\lambda_G = 100$. The smaller images on the right show similar plots for the GraphNet (GN), Robust GraphNet (RGN), Adaptive Robust GraphNet (ARGN), and Support Vector GraphNet (SVGN) classifiers at four values of the graph G diagonal scale λ_2 . Note the different scale on the ARGN models. It is of some interest that all the plots are rather slowly varying in the parameters and demonstrate significantly unimodal peaks (neither of these need be the case).

1252 VOI-based approaches, but using GraphNet classifiers on whole-brain
 1253 data, we achieve test rates from 73.0 to 74.5% for LSSO cross validation
 1254 and 71.3–73.8% for LOSO cross validation. Further, out-of-sample (OOS)
 1255 rates for the GraphNet classifiers were 67.1% (LSSO) and 69.8% (LOSO).
 1256 Thus, in this case, even out-of-sample rates with GraphNet classifiers
 1257 outperform in-sample cross validation test rates on VOI-based classi-
 1258 fiers—a considerable improvement. In taking classification accuracy as
 1259 a measure of goodness-of-fit, this indicates that GraphNet classifiers re-
 1260 sult in better fits and improved generalization relative to VOI methods,
 1261 and suggests that the resulting coefficients are a good representation of
 1262 invariant features that discriminate between choosing to purchase or
 1263 not across subjects and products.

1264 Turning to examine the coefficients, we see that the GraphNet classi-
 1265 fiers reassuringly deliver findings consistent with prior volume of in-
 1266 terest based results (Grosenick et al., 2008; Knutson et al., 2007),
 1267 replicating the observation that nucleus accumbens (NAcc) activation
 1268 begins to predict purchase choices during product presentation while
 1269 medial prefrontal cortical (MPFC) activation begins to predict pur-
 1270 chase choices during price presentation. It is also interesting to note
 1271 areas that were not included by previously applied methods, and
 1272 might not have been noticed if not for the whole-brain analysis (and
 1273 which might help account for the improved classification rates over
 1274 previous VOI analyses).

1275 While one account posits that in the context of fMRI, NAcc activation
 1276 indexes gain predictions (Knutson and Greer, 2008; Knutson et al.,
 1277 2001), an alternative account posits that NAcc activation instead index-
 1278 es gain prediction errors (e.g., Hare et al., 2008). To the extent that gain
 1279 predictions forecast future events while gain prediction errors are ad-
 1280 justments of those forecasts after an error is detected, the gain predic-
 1281 tion account posits that NAcc activation in response to products
 1282 should predict subsequent purchase choices. Applied to SHOP task
 1283 data, the robust and adaptive robust GraphNet classifier results clearly
 1284 support the gain prediction functional account of NAcc activity, since
 1285 NAcc activation in response to products predicts future choices to pur-
 1286 chase, whereas MPFC activity does not. Instead, MPFC activity predicts
 1287 choice in response to later presented price information, consistent
 1288 with a value integration account (Knutson et al., 2005; Fig. 5). The
 1289 GraphNet classifiers also revealed a previously unnoticed result in
 1290 which anterior and posterior cingulate activity clearly predicts purchase
 1291 choices at price presentation (Fig. 5). Accounts of cingulate function in
 1292 the context of purchasing remain less developed than similar accounts
 1293 of NAcc and MPFC function. Nonetheless, this result might be consistent
 1294 with attentional and salience-based accounts of posterior cingulate

function (McCoy et al., 2003), and highlights a region that deserves fur-
 1295 ther investigation in the context of choice prediction. 1296

1297 Future directions

1298 GraphNet methods can be further optimized, opening new ave-
 1299 nues for exploration. For instance, investigators might compare
 1300 graph constraints other than those related to just spatial-temporal ad-
 1301 jacency, including (1) weighted graphs derived from the data to adapt
 1302 to local smoothness, (2) cut-graphs derived from segmented brain
 1303 atlases that allow adjacent but functionally distinct regions to be in-
 1304 dependently penalized, and (3) weighted graphs derived from structural
 1305 data, which would allow constraints on voxels adjacent on a connec-
 1306 tivity graph, rather than in space or time (see Ng et al. (2012) for a
 1307 promising step in this direction). Further, investigators might use
 1308 the goodness-of-fit measure provided by GraphNet to infer which of
 1309 a set of structural graphs best relates to functional data, or to adaptive-
 1310 ly alter graph weights to explore structure in functional data (in a Var-
 1311 iational Bayes framework, for example).

1312 All of the methods considered above assume linear relationships
 1313 between input features and target variables. While this assumption
 1314 suffices in many cases, signal saturation effects alone suggest that it
 1315 might not faithfully mirror underlying physiological signals. Nonlinear
 1316 methods based on scatterplot smoothers have recently been devel-
 1317 oped and shown to work well in combination with coordinate-wise
 1318 methods (Ravikumar et al., 2009), and previous work applying sparse
 1319 regression to features derived using factor analysis have yielded
 1320 promising results (Grosenick et al., 2009a; Wager et al., 2011). In-
 1321 vestigators might thus combine nonlinear methods with sparse struc-
 1322 tured feature selection methods (Allen et al., 2011) to generate more
 1323 flexible and accurate, yet still interpretable, models of brain dynamics.
 1324 Finally, we note that because we are operating directly on voxels data,
 1325 we are working in the “native” reconstructed 3D data space rather
 1326 than on factors derived from this data or on a dictionary of basis func-
 1327 tions that approximate features of the data (e.g., wavelets). Certainly,
 1328 the optimization scheme described here would also extend to solving
 1329 problems using features derived from the data, and it is an interesting
 1330 direction for future research to explore GraphNet penalties in these
 1331 other contexts and to compare GraphNet methods to existing regres-
 1332 sion and classification methods that operate on lower dimension-
 1333 al embeddings or dictionary representations of the data. Whether
 1334 operating directly on the data with sparse structured methods or on
 1335 derived features is more appropriate will depend on the application.

1336 The methods presented here demonstrate that the former approach
 1337 can be quite effective, and provides results that are easily interpreted
 1338 in the native data space.

1339 **Appendix A. Robust GraphNet: coordinate-wise coefficient**
 1340 **updates using infimal convolution**

1341 **Algorithm 1.** Robust GraphNet update using infimal convolution

1343 1. Given a set of data and parameters $\Omega = \{X, y, \lambda_1, \lambda_G\}$, previous coeffi-
 1344 cient estimates $\hat{\alpha}^{(r)}, \hat{\beta}^{(r)}$, and $p \times p$ positive semidefinite constraint
 1345 graph $G \in S_+^{p \times p}$, let

$$\hat{\gamma}^{(r)} = [\hat{\beta}^{(r)} \hat{\alpha}^{(r)}]^T$$

$$Z = [X \ I_{n \times n}].$$

1346 2. Choose coordinate j using essentially cyclic rule (Tseng, 2001) and
 1348 fix $\tilde{\gamma} = \{\gamma_k^{(r)} | k \neq j\}$, $\tilde{Z} = Z_{\cdot, \neq j}$, $\tilde{\beta} = \{\beta_k^{(r)} | k \neq j\}$, $\tilde{X} = X_{\cdot, \neq j}$.
 1349 3. Update $\hat{\gamma}_j^{(r+1)}$ using

$$\hat{\gamma}_j^{(r+1)} \leftarrow \begin{cases} \frac{S(Z_j^T (y - \tilde{Z} \tilde{\gamma}) - (\lambda_2/2) \tilde{\gamma}^T (G_{\neq j})_{\cdot j}, \lambda_1/2)}{Z_j^T Z_j + \lambda_G G_{jj}} & \text{if } j \in \{1, \dots, p\} \\ S((y - \tilde{Z} \tilde{\gamma})_j, \lambda_1/2) & \text{if } j \in \{p+1, \dots, p+n\}, \end{cases}$$

1352 where $S(x, \lambda)$ is the element-wise soft-thresholding operator in
 1353 Eq. (30). For adaptive version replace λ_1 with $\lambda_1^* \hat{w}_j$ in above update
 1354 (see the section “Adaptive GraphNet regression”).

1355 4. Repeat steps (1)–(3) cyclically for all $j \in \{1, \dots, p+n\}$ until conver-
 1356 gence (see discussion of convergence in Friedman et al. (2007a)).
 1357 5. Optional: rescale resulting estimates using method from the sec-
 1358 tion “Rescaling coefficients to account for “double shrinking”.

1359 Derivation of updates in Algorithm 1
 1360 For a particular coordinate j , we are interested in the estimates

$$\hat{\gamma}_j = \arg \min_{\gamma_j} (1/2) \|y - \tilde{Z} \tilde{\gamma} - Z_j \gamma_j\|_2^2 + \lambda_G (\tilde{\gamma}^T (G_{\neq j})_{\cdot j} \gamma_j + G_{jj} \gamma_j^2)$$

$$+ \lambda_1 |\gamma_j| \text{ if } j \in \{1, \dots, p\},$$

$$\hat{\gamma}_j = \arg \min_{\gamma_j} (1/2) \|y - \tilde{Z} \tilde{\gamma} - Z_j \gamma_j\|_2^2 + \delta |\gamma_j| \text{ if } j \in \{p+1, \dots, p+n\}.$$

1362 By the arguments in the section “Coordinate-wise descent and
 1363 active set methods”, this yields the coordinate-wise updates

$$\hat{\gamma}_j \leftarrow \frac{S(Z_j^T (y - \tilde{Z} \tilde{\gamma}) - (\lambda_2/2) \tilde{\gamma}^T (G_{\neq j})_{\cdot j}, \lambda_1/2)}{Z_j^T Z_j + \lambda_G G_{jj}} \text{ if } j \in \{1, \dots, p\},$$

$$\hat{\gamma}_j \leftarrow \frac{S(Z_j^T (y - \tilde{Z} \tilde{\gamma}), \lambda_1/2)}{Z_j^T Z_j} \text{ if } j \in \{p+1, \dots, p+n\},$$

1366 where $S(x, \lambda)$ is the element-wise soft-thresholding operator in Eq. (30).
 1367 SVM-GraphNet classification: coordinate-wise coefficient updates
 1368 using infimal convolution.

1369 **Algorithm 2.** SVM GraphNet classification update using infimal
 1370 convolution

1371 1. Given a set of data and parameters $\Omega = \{X, y, \lambda_1, \lambda_G\}$, previous coeffi-
 1372 cient estimates $\hat{\alpha}^{(r)}, \hat{\beta}_0^{(r)}, \hat{\beta}^{(r)}$, and $p \times p$ positive semidefinite con-
 1373 straint graph $G \in S_+^{p \times p}$, let

$$\hat{\gamma}^{(r)} = [\hat{\beta}_0^{(r)} \hat{\beta}^{(r)} \hat{\alpha}^{(r)}]^T$$

$$Z = [y^T [1_{n \times 1} X] I_{n \times n}].$$

1374

2. Choose coordinate j using essentially cyclic rule (Tseng, 2001) and
 fix $\tilde{\gamma} = \{\gamma_k^{(r)} | k \neq j\}$, $\tilde{Z} = Z_{\cdot, \neq j}$, $\tilde{\beta} = \{\beta_k^{(r)} | k \neq j\}$, $\tilde{X} = X_{\cdot, \neq j}$.
 3. Update $\hat{\gamma}_j^{(r+1)}$ using

$$\hat{\gamma}_j^{(r+1)} \leftarrow \begin{cases} \tilde{\gamma}^T \tilde{Z} 1_{n \times 1} + N(\gamma_j - 1) & \text{if } j = 0 \\ S\left(\frac{\tilde{\gamma}^T \tilde{Z} 1_{n \times 1} + N(\gamma_j - 1)}{X_j^T X_j + \lambda_G G_{jj}}\right) & \text{if } j \in \{1, \dots, p\} \\ H\left(\frac{\tilde{\gamma}^T \tilde{Z} 1_{n \times 1} + N(\gamma_j - 1)}{e_j^T e_j} - 1, \delta\right) & \text{if } j \in \{p+1, \dots, p+n\}, \end{cases}$$

where $S(x, \lambda)$ is the element-wise soft-thresholding operator and
 $H(x, \delta)$ is given in Eq. (32).

4. Repeat (1)–(3) cyclically for all $j \in \{1, \dots, p+n\}$ until convergence
 (see discussion of convergence in Friedman et al. (2007a)).
 5. Optional: rescale resulting estimates using method from the sec-
 tion “Rescaling coefficients to account for “double shrinking”.

Derivation of updates in Algorithm 2

Following the description of the SVM given in the section “Turning
 regression methods into classifiers: relating Support Vector Machines
 (SVM) to penalized regression”, we can take the same approach used
 to derive the Robust GraphNet estimates with the Support Vector
 GraphNet estimates in the section “Huberized Support Vector Machine
 (SVM) GraphNet for classifications”, which we can write as

$$\hat{\gamma} = \arg \min_{\gamma} (1/2\delta) \|1_{n \times 1} - Z\gamma\|_2^2 + \lambda_G \gamma^T G \gamma$$

$$+ \sum_{j=0}^p w_j |\gamma_j| + \sum_{j=p+1}^{p+n} w_j \max(0, \gamma_j)$$

where $Z = [y^T [1_{n \times 1} X] I_{n \times n}]$, $\gamma = [\beta_0 \beta \alpha]$, $w_j = \begin{cases} 0 & \text{if } j = 0 \\ \lambda_1 & \text{if } j = 1, \dots, p \\ 1 & \text{if } j = p+1, \dots, p+n \end{cases}$

$$G' = \begin{bmatrix} 0 & 0_{1 \times p} & 0_{1 \times n} \\ 0_{p \times 1} & G & 0_{p \times n} \\ 0_{n \times 1} & 0_{n \times 1} & 0_{n \times n} \end{bmatrix} \in S_+^{(p+n+1) \times (p+n+1)}.$$

During coordinate wise descent, only one of the separable penalty
 functions has an “active” variable per descent step. Letting $h(\gamma_j) =$
 $\max(0, \gamma_j)$, we thus have

$$\hat{\gamma}_j = \begin{cases} \arg \min_{\gamma_j} (1/2\delta) \|1_{N \times 1} - \tilde{Z} \tilde{\gamma} - Z_j \gamma_j\|_2^2 & \text{if } j = 0 \\ \arg \min_{\gamma_j} (1/2\delta) \|1_{N \times 1} - \tilde{Z} \tilde{\gamma} - Z_j \gamma_j\|_2^2 + \lambda_G (\tilde{\gamma}^T (G_{\neq j})_{\cdot j} \gamma_j + G_{jj} \gamma_j^2) + \lambda_1 |\gamma_j| & \text{if } j \in \{1, \dots, p\} \\ \arg \min_{\gamma_j} (1/2\delta) \|1_{N \times 1} - \tilde{Z} \tilde{\gamma} - Z_j \gamma_j\|_2^2 + h(\gamma_j) & \text{if } j \in \{p+1, \dots, p+n\}. \end{cases}$$

Then since

$$Z_j = \begin{cases} 1_{N \times 1} & \text{if } j = 0 \\ X_j & \text{if } j \in \{1, \dots, p\} \\ e_j & \text{if } j \in \{p+1, \dots, p+n\}, \end{cases}$$

(where e_j is the vector of all zeros except for the j th element, which is
 1) we have

$$\hat{\gamma}_j \leftarrow \begin{cases} -1_{N \times 1}^T Z_j + (\tilde{Z} \tilde{\gamma})^T Z_j + \gamma_j Z_j^T Z_j & \text{if } j = 0 \\ S\left(\frac{(\tilde{Z} \tilde{\gamma})^T Z_j - 1_{N \times 1}^T Z_j - (\lambda_G/2) \tilde{\gamma}^T (G_{\neq j})_{\cdot j}, \lambda_1/2)}{Z_j^T Z_j + \lambda_G G_{jj}}\right) & \text{if } j \in \{1, \dots, p\} \\ \frac{H\left(\frac{(\tilde{Z} \tilde{\gamma})^T e_j - 1_{N \times 1}^T e_j, \delta}{e_j^T e_j}\right)}{e_j^T e_j} & \text{if } j \in \{p+1, \dots, p+n\}, \end{cases}$$

1406 yielding update:

$$\hat{\gamma}_j \leftarrow \begin{cases} \tilde{\gamma}^T \tilde{Z}_{N \times 1} + N(\gamma_j - 1) & \text{if } j = 0 \\ S \left(\frac{\left(\tilde{Z}^T \tilde{\gamma} - 1_{N \times 1} \right)^T X_{j-} - (\lambda_G/2) \beta^T (G_{\neq j})_{\cdot j} \lambda_1/2}{X_{j-}^T X_{j-} + \lambda_G G_{jj}} \right) & \text{if } j \in \{1, \dots, p\} \\ H \left(\frac{\left(\tilde{Z} \tilde{\gamma} \right)_j - 1}{\delta} \right) & \text{if } j \in \{p+1, \dots, p+n\}, \end{cases}$$

1408 where $S(x, \lambda)$ is the element-wise soft-thresholding operator in Eq. (30)
1410 and

$$H(x, \delta) = \begin{cases} x - \delta & \text{if } x < 1 \\ x & \text{otherwise.} \end{cases} \quad (32)$$

1412

Parameter grid used in cross-validation

1414 Parameters $\{\lambda_1, G, \lambda_G, \delta, \lambda_1^*\}$ were taken over the following grid of
1415 values:

$$\lambda_1 \in \{10, 11, \dots, 99\}$$

$$G \in \{L, L + \eta l, L + 10^2 \eta l, L + 10^3 \eta l, L + 10^4 \eta l\} \text{ where } \eta = 1/\lambda_G \text{ for } \lambda_G > 0 \text{ and } 1 \text{ otherwise}$$

$$\lambda_G \in \{0, 10^1, 10^2, 10^3, 10^4, 10^5\}$$

$$\delta \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1, 2, 10, 100\}$$

$$\lambda_1^* \in \{1, 10^{-1}, 10^{-2}\}.$$

1416

The linear SVM was fit over parameters

1418

$$C \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 2, 3, 4, 5, 6, 7, 10^1, 10^2, 10^3\}. \quad (33)$$

1420

1421 References

- 1422 Adler, R.J., Taylor, J.E., 2000. Random Fields and Geometry.
1423 Allen, G., Grosenick, L., Taylor, J.E., 2011. A generalized least squares matrix decomposition.
1424 Rice University Technical Report No. TR2011-03 (Preprint available on arXiv:1102.3074).
1425 Barbieri, M., Berger, J., 2004. Optimal predictive model selection. *Ann. Stat.* 32, 870–897.
1426 Belkin, M., Niyogi, P., 2008. Towards a theoretical foundation for Laplacian-based manifold methods. *J. Comput. Syst. Sci.* 74, 1289–1308.
1427 Belkin, M., Niyogi, P., Sindhvani, V., 2006. On manifold regularization. *J. Mach. Learn. Res.* 7, 2399–2434.
1430 Boyd, S., Vandenberghe, L., 2004. *Convex Optimization*.
1431 Bray, S., Chang, C., Hoeff, F., 2009. Applications of multivariate pattern classification analyses in developmental neuroimaging of healthy and clinical populations. *Front. Hum. Neurosci.* 3 (32), 1–12.
1434 Breiman, L., Iha, R., 1984. Univ of California at Berkeley Technical Report: Nonlinear Discriminant Analysis via Scaling and ACE.
1436 Brodersen, K., Haiss, F., Ong, C.S., Jung, F., Tittgemeyer, M., Buhmann, J.M., Weber, B., Stephan, K.E., 2011. Model-based feature construction for multivariate decoding. *NeuroImage* 56 (2), 601–615.
1439 Candes, E.J., Wakin, M.B., Boyd, S.P., 2003. Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *PNAS* 100 (5), 2197–2202.
1440 Candes, E.J., Wakin, M.B., Boyd, S.P., 2008. Enhancing sparsity by reweighted l_1 minimization. *J. Fourier Anal. Appl.* 14, 69–82.
1443 Carroll, M., Cecchi, G., Rish, I., Garg, R., 2009. Prediction and interpretation of distributed neural activity with sparse models. *NeuroImage* 44 (1), 112–122.
1445 Chappell, M.A., Groves, A.R., Whitcher, B., Woolrich, M.W., 2009. Variational Bayesian inference for a nonlinear forward model. *IEEE Trans. Signal Proc.* 57 (1), 223–236.
1446 Clemmensen, L., Hastie, T., Witten, D., Ersoy, B., 2011. Sparse discriminant analysis. *Technometrics* 53 (4), 406–413.
1448 Cortes, C., Vapnik, V., 1995. Support-vector networks. *Mach. Learn.* 20 (3), 273–297.
1450 Cox, R.W., 1996. AFNI: software for analysis and visualization of functional magnetic resonance images. *Comput. Biomed. Res.* 29, 162–173.
1452 De Martino, F., Gentile, F., Esposito, F., Balsi, M., Di Salle, F., Goebel, R., Formisano, E., 2007. Classification of fMRI independent components using IC-fingerprints and support vector machine classifiers. *NeuroImage* 34, 177–194.
1455 der Kooij, A.V., 2007. Prediction accuracy and stability of regression with optimal scaling transformations. Technical Report, Dept. Data Theory, Leiden Univ.
1457 Donoho, D.L., 1995. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory* 41 (3), 613–627.
1458 Donoho, D., 2006. For most large underdetermined systems of linear equations, the minimal l_1 -norm solution is also the sparsest solution. *Commun. Pure Appl. Math.* 59 (6), 782–797.

- Etzel, J.A., Gazzola, V., Keysers, C., 2009. An introduction to anatomical ROI-based fMRI classification analysis. *Brain Res.* 1282, 114–125.
1462
1463 Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Am. Stat. Assoc.* 96 (456), 1348–1360.
1464
1465 Friedman, J.H., 1997. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Min. Knowl. Disc.* 1, 55–77.
1466
1467 Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., 2007a. Pathwise coordinate optimization. *Ann. Appl. Stat.* 1 (2), 302–332.
1468
1469 Friedman, J.H., Hastie, T., Höfling, H., Tibshirani, R., 2007b. Pathwise coordinate optimization. *Ann. Appl. Stat.* 1 (2), 302–332.
1470
1471 Friedman, J., Hastie, T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.* 33 (1), 1–22.
1472
1473 Friston, K., Holmes, A., Worsley, K., Poline, J., 1995. Statistical parametric maps in functional imaging: a general linear approach. *Hum. Brain Mapp.* 2, 189–210.
1474
1475 Glover, G.H., Law, C.S., 2001. Spiral in/out BOLD fMRI for increased SNR and reduced susceptibility artifacts. *Magn. Reson. Med.* 46, 512–522.
1476
1477 Grosenick, L., Greer, S.M., Knutson, B., 2008. Interpretable classifiers for fMRI improve prediction of purchases. *IEEE Trans. Neural Syst. Rehabil. Eng.* 16 (6), 539–548.
1478
1479 Grosenick, L., Anderson, T., Smith, S.J., 2009a. Elastic source selection for in vivo imaging of neuronal ensembles. *Biomedical Imaging: From Nano to Macro*, 6th IEEE International Symposium on.
1480
1481 Grosenick, L., Klingenberg, B., Greer, S., Taylor, J.E., Knutson, B., 2009b. Whole-brain sparse penalized discriminant analysis for predicting choice. *NeuroImage (Supplement 1)*, S58.
1482
1483 Grosenick, L., Klingenberg, B., Knutson, B., Taylor, J.E., 2010. Interpretable multivariate models for whole-brain fMRI. *NeuroImage (OHBM 2010)*.
1484
1485 Q15 Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46, 389–422.
1486
1487 Hanke, M., Halchenko, Y., Sederberg, P., 2009. PyMVPA: a Python toolbox for multivariate pattern analysis of fMRI data. *Neuroinformatics* 7 (1), 37–53.
1488
1489 Hare, E., O'Doherty, J., Camerer, C.F., Schultz, W., Rangel, A., 2008. Dissociating the role of the orbitofrontal cortex and the striatum in the computation of goal values and prediction errors. *J. Neurosci.* 28 (22), 5623–5630.
1490
1491 Hastie, T., Tibshirani, R., Buja, A., 1994. Flexible Discriminant Analysis by Optimal Scoring. *J. Am. Stat. Assoc.* 89 (428), 1255–1270.
1492
1493 Hastie, T., Buja, A., Tibshirani, R., 1995. Penalized Discriminant Analysis. *Ann. Stat.* 23 (1), 73–102.
1494
1495 Hastie, T., Tibshirani, R., Friedman, J., 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*.
1496
1497 Haynes, J., Rees, G., 2006. Decoding mental states from brain activity in humans. *Nat. Rev. Neurosci.* 7, 523–534.
1498
1499 Haynes, J., Sakai, K., Rees, G., Gilbert, S., 2007. Reading hidden intentions in the human brain. *Curr. Biol.* 17 (4), 323–328.
1500
1501 Hoerl, A.E., Kennard, R.W., 1970. Ridge regression: applications to nonorthogonal problems. *Technometrics* 12 (1), 69–82.
1502
1503 Huber, P.J., Ronchetti, E.M., 2009. *Robust Statistics*.
1504
1505 Hutchinson, R.A., Niculescu, R.S., Keller, T.A., Rustandi, I., Mitchell, T.M., 2009. Modeling fMRI data generated by overlapping cognitive processes with unknown onsets using Hidden Process Models. *NeuroImage* 46 (1), 87–104.
1506
1507 Jenatton, R., Audibert, J.-Y., Bach, F., 2011. Structured variable selection with sparsity-inducing norms. *J. Mach. Learn. Res.* 12, 2777–2824.
1508
1509 Jimenez, A.B., Lazaro, J.L., Dorronsoro, J.R., 2009. Finding optimal model parameters by deterministic and annealed focused grid search. *Neurocomputing* 72 (13–15), 2824–2832.
1510
1511 Karmarkar, U. R., Shiv, B., Knutson, B., submitted for publication. Sticker shock: the neural and behavioral impact of price primacy on purchasing.
1512
1513 Knutson, B., Greer, S.M., 2008. Anticipatory affect: neural correlates and consequences for choice. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 363 (1511), 3771–3786.
1514
1515 Knutson, B., Adams, C.M., Fong, G.W., Hommer, D., 2001. Anticipation of increasing monetary reward selectively recruits nucleus accumbens. *J. Neurosci.* 21 (16), 1–5.
1516
1517 Knutson, B., Taylor, J., Kaufman, M., Peterson, R., Glover, G., 2005. Distributed neural representation of expected value. *J. Neurosci.* 25, 4806–4812.
1518
1519 Knutson, B., Rick, S., Wimmer, G.E., Prelec, D., Loewenstein, G., 2007. Neural predictors of purchases. *Neuron* 53, 147–156.
1520
1521 Lehmann, E.L., Casella, G., 1998. *Theory of Point Estimation*.
1522
1523 Teng, C., 2008. Sparse optimal scoring for multiclass cancer diagnosis and biomarker detection using microarray data. *Comput. Biol. Chem.* 32, 417–425.
1524
1525 Li, C., Li, H., 2008. Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics* 24 (9), 1175–1182.
1526
1527 Li, Y., Namburi, P., Yu, Z., Guan, C., Feng, J., Gu, Z., 2009. Voxel selection in fMRI data analysis based on a sparse representation. *IEEE Trans. Biomed. Eng.* 56 (10), 2439–2451.
1528
1529 McCoy, A.N., Crowley, J.C., Haghigian, G., Dean, H.L., Platt, M.L., 2003. Saccade reward signals in posterior cingulate cortex. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 40 (5), 1031–1040.
1530
1531 Meinshausen, N., Bühlmann, P., 2010. Stability Selection. *J. R. Stat. Soc. Ser. B* 72 (4), 417–473.
1532
1533 Merchante, L.F.S., Grandvalet, Y., Govaert, G., 2012. An efficient approach to sparse linear discriminant analysis. *Proceedings of the International Conference on Machine Learning (ICML)*.
1534
1535 Michel, V., Gramfort, A., Varoquaux, G., Eger, E., Thirion, B., 2011. Total variation regularization for fMRI-based prediction of behavior. *IEEE Trans. Med. Imaging* 30 (7), 1328–1340.
1536
1537 Mitchell, T.M., Hutchinson, R., Niculescu, R., 2004. Learning to decode cognitive states from brain images. *Mach. Learn.* 57 (1–2), 145–175.
1538
1539 Mohamed, S., Heller, K., Ghahramani, Z., 2011. Bayesian and L1 Approaches to Sparse Unsupervised Learning (arXiv:1106.1157v2).
1540
1541
1542
1543
1544
1545
1546

- 1547 Mourão-Miranda, J., Friston, K., Brammer, M., 2007. Dynamic discrimination analysis: a
1548 spatial-temporal SVM. *NeuroImage* 36, 88–99. 1586
- 1549 Ng, B., Siless, V., Varoquaux, G., Yoline, J.-B., Thirion, B., Abugharbieh, R., 2012. Connec-
1550 tivity-informed sparse classifiers for fMRI brain decoding. *Pattern Recognition in*
1551 *NeuroImaging (PRNI), IEEE 2012 International Workshop on*, pp. 101–104. 1588
- 1552 Norman, K., Polyn, S., Detre, G., Haxby, J., 2006. Beyond mind-reading: multi-voxel pat-
1553 tern analysis of fMRI data. *Trends Cogn. Sci.* 10 (9), 424–430. 1589
- 1554 O’Toole, A., Jiang, F., Abdi, H., Penard, N., 2007. Theoretical, statistical, and practical per-
1555 spectives on pattern-based classification. *J. Cogn. Neurosci.* 2007 (19), 1735–1752. 1590
- 1556 Peelen, M., Fei-Fei, L., Kastner, S., 2009. Neural mechanisms of rapid natural scene cat-
1557 egorization in human visual cortex. *Nature* 460, 94–97. 1591
- 1558 Pereira, F., Mitchell, T.M., Botvinick, M., 2009. Machine learning classifiers and fMRI: a
1559 tutorial overview. *NeuroImage* 45 (1), S199–S209 (Supplement). 1592
- 1560 Polyn, S., Natu, V., Cohen, J., Norman, K., 2005. Category-specific cortical activity pre-
1561 cedes retrieval during memory search. *Science* 310 (5756), 1963–1966. 1593
- 1562 Ravikumar, P., Vu, V.Q., Yu, B., Naselaris, T., Kay, K.N., Gallant, J.L., 2009. Nonparametric
1563 sparse hierarchical models describe V_1 fMRI responses to natural images: Ad-
1564 vances in *Neural Information Processing Systems*, 21. 1594
- 1565 Rockafellar, T.J., 1970. *Convex Analysis*. 1595
- 1566 Rudin, L.I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal al-
1567 gorithms. *Physica D* 50, 259–268. 1596
- 1568 Ryali, S., Supekar, K., Abrams, D.A., Menon, V., 2010. Sparse logistic regression for
1569 whole-brain classification of fMRI data. *NeuroImage* 51 (2), 752–764. 1597
- 1570 Shinkareva, S.V., Mason, R.A., Malave, V.L., Wang, W., Mitchell, T.M., Just, M.A., 2008.
1571 Using fMRI brain activation to identify cognitive states associated with perception
1572 of tools and dwellings. *PLoS One* 3 (1). 1598
- 1573 Slawski, M., zu Castell, W., Tutz, G., 2010. Feature selection guided by structural infor-
1574 mation. *Ann. Appl. Stat.* 4 (2), 1055–1080. 1599
- 1575 Stein, C., 1956. Inadmissibility of the usual estimator for the mean of a multivariate normal
1576 distribution. *Proc. Third Berkeley Symp. on Math. Statist. and Prob.*, vol. 1, pp. 197–206. 1600
- 1577 Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser.*
1578 *B* 58 (1), 267–288. 1601
- 1579 Tibshirani, R., Taylor, J.E., 2011. The solution path of the generalized lasso. *Ann. Stat.* 3,
1580 1335–1371. 1602
- 1581 Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K., 2005. Sparsity and smoothness
1582 via the fused lasso. *J. R. Stat. Soc. Ser. B* 67 (1), 91–108. 1603
- 1583 Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J.E., Tibshirani, R.J., 2012.
1584 Strong rules for discarding predictors in lasso-type problems. *J. R. Stat. Soc. Ser. B*
1585 *74* (2), 245–266. 1604
- 1625 1625
- Tikhonov, A., 1943. On the Stability of Inverse Problems. 1586
- Tseng, P., 1988. Technical Report LIDS-P, 1840. Massachusetts Institute of Technology, 1587
Laboratory for Information and Decision Systems (arXiv:0903.2515). 1588
- Tseng, P., 2001. Convergence of block coordinate descent method for nondifferentiable
maximization. *J. Optim. Theory Appl.* 109, 474–494. 1589
- van Gerven, M.A.J., Heskes, T., 2012. A linear Gaussian framework for decoding of per-
ceived images. *International Workshop on Pattern Recognition in NeuroImaging*
(PRNI), IEEE 2012, pp. 1–4. 1591
- van Gerven, M.A.J., Cseke, B., de Lange, F.P., Heskes, T., 2010. Efficient Bayesian mul-
tivariate fMRI analysis using a sparsifying spatio-temporal prior. *NeuroImage*
150–161. 1594
- Wager, T.D., Atlas, L.Y., Leotti, L.A., Rillings, J.K., 2011. Predicting individual differences
in placebo analgesia: contributions of brain activity during anticipation and pain
experience. *J. Neurosci.* 31 (2), 439–452. 1598
- Wang, L., Zhu, J., Zou, H., 2008a. Hybrid Huberized Support Vector Machines
for microarray classification and gene selection. *Bioinformatics* 24 (3),
412–419. 1600
- Wang, Y., Junfeng, Y., Yin, W., Zhang, Y., 2008b. A new alternating minimization algo-
rithm for Total Variation image reconstruction. *SIAM J. Imaging Sci.* 1 (3),
248–272. 1603
- Wipf, D., Nagarajan, S., 2008. A new view of automatic relevance determination: Ad-
vances in *Neural Information Processing Systems*, 20. 1606
- Witten, D.M., Tibshirani, R., 2011. Penalized classification using Fisher’s linear discrim-
inant. *J. R. Stat. Soc. Ser. B* 73 (5), 753–772. 1607
- Worsley, K.J., Taylor, J.E., Tomaiuolo, F., Lerch, J., 2004. Unified univariate and multi-
variate random field theory. *NeuroImage* 23, S189–S195. 1609
- Yamashita, O., Sato, M., Yoshioka, T., Tong, F., Kamitani, Y., 2008. Sparse estimation au-
tomatically selects voxels relevant for the decoding of fMRI activity patterns.
NeuroImage 42 (4), 1414–1429. 1611
- Zhou, S., van de Geer, S., Bühlmann, P., 2011. Adaptive lasso for high dimensional re-
gression and Gaussian graphical modeling. *Electron. J. Stat.* 5, 688–749. 1612
- Zou, H., 2006. The adaptive lasso and its oracle properties. *J. Am. Stat. Assoc.* 101 (467),
1418–1429. 1613
- Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. *J. R.*
Stat. Soc. Ser. B 67 (2), 301–320. 1614
- Zou, H., Zhang, H., 2009. On the adaptive elastic-net with a diverging number of param-
eters. *Ann. Stat.* 37 (4), 1733–1751. 1615
- Zou, H., Hastie, T., Tibshirani, R., 2007. On the “degrees of freedom” of the lasso. *Ann.*
Stat. 35 (5), 2173–2192. 1616