Interpretable Whole-Brain Prediction Analysis with GraphNet.

Article *in* NeuroImage · January 2013

DOI: 10.1016/j.neuroimage.2012.12.062 · Source: PubMed

Logan Grosenick
Columbia University

Brian Knutson
Stanford University

A family of interpretable multivariate models for regression and classification of whole-brain fMRI data

Logan Grosenick[a,1,*], Brad Klingenberg[b], 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] *Deparment of Psychology, Stanford University, Stanford, CA, USA*

Abstract

Increasing interest in applying statistical learning methods to functional magnetic resonance imaging (fMRI) data has led to growing application of off-the-shelf classification and regression methods. These methods allow investigators to use standard software packages to accurately decode perceptual, cognitive, or behavioral states from distributed patterns of neural activity. However, such models are generally difficult to interpret when fit to whole-brain fMRI data, as they suffer from coefficient instability, are sensitive to outliers, and typically return dense rather than parsimonious solutions. Here, we develop several variants of the the graph-constrained elastic net (GraphNet)–a fast, whole-brain regression and classification method developed for spatial and temporally correlated data that yields interpretable model parameters (Grosenick et al. 2009). GraphNet models yield interpretable solutions by incorporating sparse graph priors based on model smoothness or connectivity as well as a global sparsity inducing prior that automatically selects important variables. Because these methods can be fit to large data sets efficiently and can improve accuracy relative to volume-of-interest (VOI) methods, they allow investigators to take a more objective approach to model fitting—avoiding the selection bias associated with VOI analyses or the multiple comparison problems of roaming VOI ("searchlight") methods. As fMRI data are unlikely to be Gaussian, we (1) extend GraphNet to include robust loss functions that confer insensitivity to outliers, (2) equip them with "adaptive" penalties that have oracle properties, and (3) develop a novel sparse structured support vector machine (SVM) GraphNet classifier to allow maximum-margin classification with GraphNet models. 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 (Knutson et al. 2007; Grosenick et al. 2008) while discovering task-related regions not included in the original VOI approach.

[*] Corresponding author
[1] Correspondence to:
Logan Grosenick
220 Panama Street, Ventura Hall Rm 30
Stanford, CA 94305
(650) 283-0010

1. Introduction

In functional magnetic resonance imaging (fMRI) studies, investigators measure neural "activity" with the blood oxygen-level dependent (BOLD) signal, and relate this activity 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, there has been a growing number of neuroimaging studies applying statistical learning to fMRI data in order 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 approachs have enabled investigators to use patterns of activation across multiple voxels to classify categories of images during subject viewing (Peelen et al. 2009; Shinkareva et al. 2008), categories of images during memory retrieval (Polyn et al. 2005), intentions to act (Haynes et al. 2007), and intentions to purchase (Grosenick et al. 2008), to name just a few applications (see also Norman et al. 2006; Haynes and Rees 2006; Pereira et al. 2009; O'Toole et al. 2007; Bray et al. 2009 and the many examples in NeuroImage Volume 56 Issue 2). In a variety of cases, statistical learning algorithms have increased model sensitivity relative to standard mass-univariate analyses (Haynes and Rees 2006; Pereira et al. 2009).

Despite these advances, the application of statistical learning to neuroimaging data is still developing. Most research applying statistical learning to fMRI has been conducted by a few labs (Norman et al. 2006), and it remains standard practice to use off-the-shelf classifiers (Norman et al. 2006; Pereira et al. 2009) (but cf. Hutchinson et al. 2009; Chappell et al. 2009; Brodersen et al. 2011). These models are typically applied to volume of interest data within subjects instead of to whole-brain data across subjects (Etzel et al. 2009; Pereira et al. 2009) (but cf. Mitchell et al. 2004; Mourão-Miranda et al. 2007; Grosenick et al. 2009; 2010; Ryali et al. 2010). While such classifiers have a venerable history in the machine learning literature, they were not developed with whole-brain neuroimaging data in mind, and so suffer from inefficiencies when applied in the neuroimaging context. In particular, the large number of features (usually voxel data) and spatiotemporal autocorrelations characteristic of fMRI data present unique problems that off-the-shelf classifiers are not designed to solve.

Indeed, in the machine learning literature, the purpose of off-the-shelf classifiers such as discriminant analysis (DA), naive Bayes (NB), k-nearest neighbors (kNN), and support vector machines (SVM) has been to quickly and easily yield high classification accuracy on tasks where accuracy itself is the primary objective–for example speech recognition or hand-written digit identification (Hastie et al. 2009). However, beyond accuracy, neuroscientists would like to make claims about what brain areas correlate with particular behaviors or stimuli at particular points in time (or more precisely, which mental properties supervene on brain properties (Vogelstein et al. In Press)). This aim requires classification or regression methods that yield clearly interpretable sets of model coefficients. For this reason, recent literature on multivariate classification of fMRI data recommends that linear classifiers such as logistic regression (LR), linear discriminant analysis (LDA), Gaussian Naive Bayes (GNB), or linear SVM be preferred over nonlinear classifiers in fMRI applications (Haynes and Rees 2006; Pereira et al. 2009).

However, it is a mistake to think that linearity alone will guarantee that a model will yield stable and interpretable coefficients. For instance, in the case of many correlated input variables, it is known that LR, LDA, and GNB all yield unstable coefficients and degenerate covariance estimates–particularly if applied to smoothed data (Hastie et al. 1995; 2009). In the classification context, penalized least squares approaches tend to oversmooth the coefficients, complicating interpretation (Friedman 1997). Additionally, most linear classifiers return dense sets of coefficients (as in Figure 1, left panels) that require a threshold or a feature selection step in order to to yield a parsimonious model. Although some heuristic methods exist for coefficient selection, these are in general greedy (e.g., forward/backward stage-wise procedures like Recursive Feature Elimination (Guyon and Vapnik 2002; De Martino et al. 2007; Bray et al. 2009)) and can be unstable over resampling of the data (as they are easily stuck in local minima) (Hastie et al. 2009). While principled

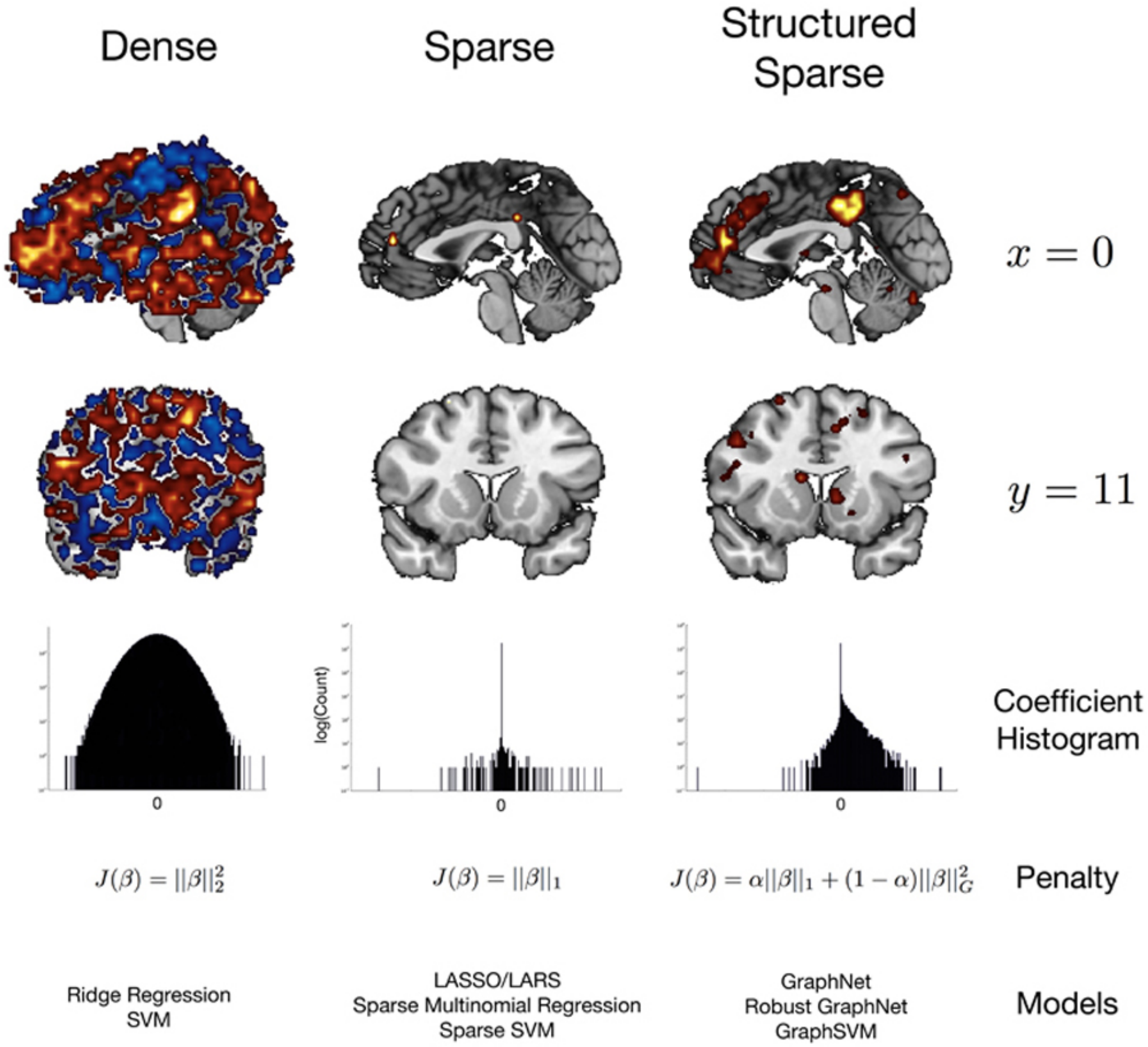

Figure 1. Mid-sagittal and coronal plots of coefficients from dense, sparse, and structured sparse models. Warm colored coefficients indicate a positive relationship with the target variable (here predicting the decision to buy a product, cool colors a negative relationship. Sparse models set many model coefficients to zero, while in dense models almost all coefficients are nonzero. Structured sparse models use a penalty on differences between selected voxels to impose a structure on the model 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 model coefficients have a near Laplacian (double-exponential) distribuion, while the dense coefficients have a near Gaussian distribution. The structured sparse model coefficients are a product of these distributions (see also Figure 2). Coefficients penalties that yield each model type and examples of such models are given below each column.

methods exist for choosing a threshold for dense mass-univariate maps of coefficients (e.g. Random Field Theory (Adler and Taylor 2000; Worsley et al. 2004)), such approaches are not currently available for dense multivariate regression and classification methods.

Recently, sparse regression models have been applied to neuroimaging data to yield reduced sets of coefficients chosen automatically during the model fitting process. The first examples in the fMRI literature are (Yamashita et al. 2008), who apply sparse logistic regression (Tibshirani 1996) to classification of visual stimuli, and (Grosenick et al. 2008) who develop sparse penalized discriminant analysis to turn "elastic net" regression (Zou and Hastie 2005) into a classifier and apply it to choice prediction. Subsequently, sparse models 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 VOIs, whole-brain volumes (Ryali et al. 2010; van Gerven et al. 2010) and whole brain volumes over multiple time points (Grosenick et al. 2009; 2010). In general these methods use an ℓ_1-penalty (sum of absolute values) on the model coefficients, which results in many of the estimated coefficients being set to zero (see Figure 1, leftmost panels, and Figure 2b). When applied without modification to correlated fMRI data, however, ℓ_1-penalized models can select an overly sparse solution–resulting in omission of relevant features and yielding unstable coefficient estimates over cross-validation folds Zou and Hastie (2005); Grosenick et al. (2008). To allow relevant but correlated coefficients to coexist in a sparse model, recent approaches to fMRI regression (Carroll et al. 2009; Li et al. 2009) and classification (Grosenick et al. 2008; 2009; Ryali et al. 2010) use a hybrid ℓ_1- and ℓ_2-norm penalty ("elastic net" penalty Zou and Hastie (2005)) on the coefficients. These approaches allow correlated variables to be included together in a sparse model.

We explore models that employ a modified version of the the elastic net penalty that includes a general user-specified sparse graph penalty. This penalty allows the user to efficiently incorporate physiological constraints and prior information in a model. The resulting graph-constrained elastic net (GraphNet) regression (Grosenick et al. 2009; 2010) can find "structured sparsity" in correlated data with many features (Figure 1, right panels) and is related to previous approaches in the manifold learning (Belkin et al. 2006) and gene microarray literatures (Li and Li 2008). Related "sparse structured" methods in the statistics literature have recently been shown to have desirable convergence and variable selection properties for large correlated data (Slawski et al. 2010; Jenatton et al. 2011), and structured sparse coefficients can also be achieved within a Bayesian framework (van Gerven et al. 2010). Here we extend the performance of GraphNet regression and classification methods on whole-brain fMRI data, making them robust to outliers (for both regression and classification), adding "adaptive" penalization to reduce model bias and improve model selection, and developing a novel support vector GraphNet (SVGN) classifier. To allow these methods to be fit to whole-brain fMRI data over multiple time-points, we adapt results from the applied statistics literature (Friedman and Tibshirani 2010) to fit them models to large data sets efficiently.

After developing robust and adaptive GraphNet regression and classification methods, we demonstrate the utility of GraphNet classifiers on previously published data (Knutson et al. 2007). Specifically, we use them to predict subjects' purchasing behavior on a trial-to-trial basis using their whole-brain BOLD response over several time points at once, and make inferences about which brain regions discriminate upcoming decisions to purchase versus not purchase a product. Fitting the models to 25 subjects whole-brain data over 7 TRs yielded classification rates better than those found previously in a volume-of-interest (VOI) based classification analysis (Grosenick et al. 2008), and than those obtained with the linear support vector machine (SVM) classifier fit to the full data. Our whole-brain, spatiotemporal results confirm the relevance of the VOIs chosen in the previous studies (bilateral nucleus accumbens, medial prefrontal cortex, and insula), but also implicate previously unchosen areas (notably bilateral ventral tegmental area (VTA) and posterior cingulate). We conclude with a discussion of how to interpret GraphNet model coefficients appropriately, as well as future improvements, applications, and extensions of these models.

2. Methods

2.1. Preliminaries

2.1.1. Penalized least squares regression

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. 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

$$y = X\beta + \epsilon, \tag{1}$$

where ϵ is a noise term typically assumed to be normally distributed $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with mean 0 and variance σ^2. In this case using squared error loss leads to the well-known ordinary least squares (OLS) solution

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \; \|y - X\beta\|_2^2 = (X^T X)^{-1} X^T y, \tag{2}$$

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 and 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 in order to stabilize them. This is accomplished by adding a penalty term $\mathcal{P}(\beta)$ on the coefficient vector β yielding

$$\widehat{\beta} = \underset{\beta}{\operatorname{argmin}} \; \|y - X\beta\|_2^2 + \lambda \mathcal{P}(\beta), \;\; \lambda \in \mathbb{R}_+, \tag{3}$$

where λ is a parameter that trades off fit with sparsity (variance with bias) and $\mathbb{R}_+$ is the set of nonnegative scalars. These estimates are equivalent to maximimum 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 section 2.3).

2.1.2. Sparse regression and automatic variable selection

There are a few standard choices for penalty $\mathcal{P}(\beta)$. Letting $\mathcal{P}(\beta) = \|\beta\|_2^2 = \sum_{j=1}^p \beta_j^2$ (the ℓ_2-norm) gives the now classical "ridge" regression estimates originally proposed for such problems (Tikhonov 1943; Hoerl and Kennard 1970). 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 exceedingly 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, Figure 1 left panels). In many applications having a sparse vector $\hat{\beta}$ is highly desirable; a model with fewer non-zero coefficients is simpler, and can help identify and emphasize which predictors 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 jth coefficient β_j is nonzero and 0 otherwise. This is just a penalty on the number of nonzero coefficients, and corresponds to finding the sparsest possible model. However, finding such a solution involves a combinatorial search through possible models (a form of "all subsets regression" (Hastie et al. 2009)) and so is computationally infeasible for even a modestly large number of input features. An ℓ_1-norm penalty can be used as a heuristic that results in coefficient sparsity (in the maximum a posteriori (MAP) estimates—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 model in many cases (Candes et al. 2003; Donoho 2006), they do not always do so, and reweighted methods such as Automatic Relevance Determination (Wipf and Nagarajan 2008) and iterative reweighting of the ℓ_1-penalty Candes et al. (2008) exist for finding sparser estimates.

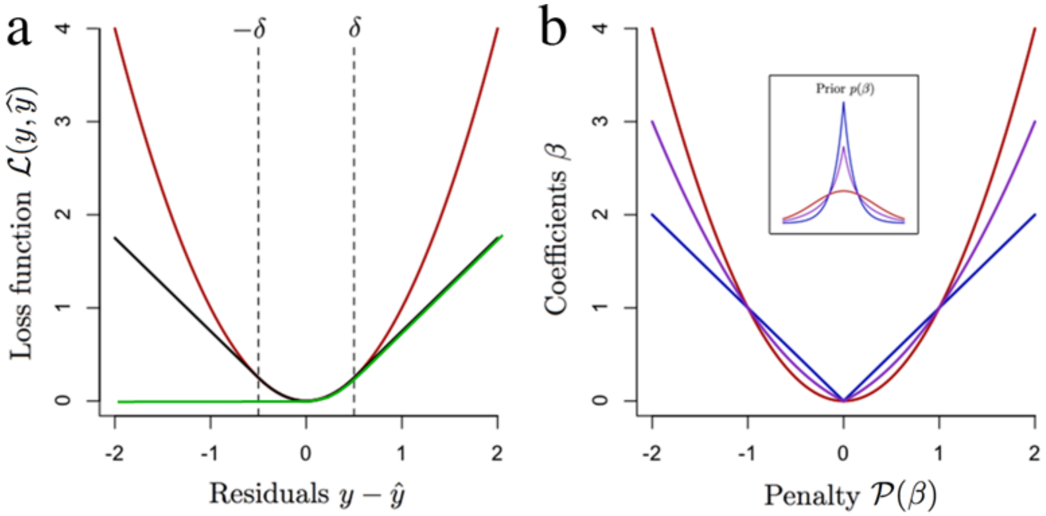

Figure 2. (a) Diagrammatic representation of squared-error (red), Huber (black), and Huberized Hinge (green) loss functions. Dotted lines denote where the Huber loss changes from penalizing residuals quadratically (where $|y - \widehat{y}| \leq \delta$) to penalizing them linearly (where $|y - \widehat{y}| > \delta$). 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 "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 read and blue curves: $\alpha\beta^2 + (1 - \alpha)|\beta|$ (where here $\alpha = 0.5$), called the "Elastic Net" penalty. The inset shows the prior distribution on the coefficient estimates that each of these penalties corresponds to: Gaussian (red), Laplacian (blue), and mixed Gaussian and Laplacian (purple).

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, if there is a group of highly correlated predictors the lasso will typically select a subset of "representative" predictors from this group to include in the model (Zou and Hastie 2005). This can make it difficult to interpret $\widehat{\beta}$ because a coefficient being 0 doesn't necessarily mean the corresponding predictor isn't useful for modeling y (i.e., false negatives are likely). Worse, entirely different subsets of the coefficients may be selected over resamplings of the data (during cross-validation, for example). Moreover, the lasso can select at most n non-zero coefficients(Zou and Hastie 2005), which may be undesirable in problems where the number of input features p is much larger than the number of observations n ("$p >> n$" problems). Finally, as a global shrinkage method it biases model coefficients towards zero (Tibshirani 1996; Hastie et al. 2009), making it difficult to interpret their magnitude in terms of original data units. Other models based on only an ℓ_1-penalty, such as sparse logistic/multinomial regression and sparse SVM (Hastie et al. 2009) suffer from the same problems.

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 regularization, and may be written

$$\widehat{\beta} = \kappa \underset{\beta}{\operatorname{argmin}} \|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2. \tag{4}$$

This elastic net estimator overcomes several (though not all) of the disadvantages discussed above, while maintaining many advantages of ridge regression and the lasso. In particular, the elastic net accomodates groups of correlated variables in the model 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 permits correlated variables in the model and stabilized the sample covariance estimate. This approach has been shown to perform well on fMRI data in both regression and classification settings (Grosenick et al. 2008; Carroll et al. 2009; Ryali et al. 2010). The factor $\kappa = 1 + \lambda_2$ in (4) and subsequent equations is a rescaling factor discussed in further detail below.

2.1.3. Optimal Scoring (OS) and Sparse Penalized Discriminant Analysis (SPDA)

Sparse regression methods like the lasso or elastic net can be turned into sparse classifiers (Leng 2008; Grosenick et al. 2008; Clemmensen et al. In Press). 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 $\widehat{y}$ and classify to 0 if the ith estimate $\widehat{y}_i < 0.5$ and to 1 if the estimate $\widehat{y}_i > 0.5$ (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 ith row of Y has a 1 in the jth column if the observation is in the jth class and 0s otherwise. If we run a multiple linear regression of Y on our predictors X, we can the results for a one-against-all classification by classifying the ith observation to the class having the largest fitted value $\widehat{Y}_{i1}, \widehat{Y}_{i2}, ..., \widehat{Y}_{iJ}$. With the exception of binary classification on balanced data, this classifier has several disadvantages, including that the estimates $\widehat{Y}_{ij}$ are not probabilities, and that in the multi-class case certain classes can be "masked" by others, resulting in decreased classification accuracy (Hastie et al. 2009). However, it has been known for some time that 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 in some cases dramatically improving the classification performance over the original multivariate regression (Hastie et al. 1994; 1995; 2009).

(Hastie et al. 1994) and (Hastie et al. 1995) exploit this fact and an equivalence 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 Appendix). This 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 (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 $\theta : g \to \mathbb{R}$ that maps classes $g \in \{1, ...J\}$ into the real numbers. In the case of a multi-class classification using the elastic net, we can apply OS to yield estimates

$$\begin{aligned} (\widehat{\Theta}, \widehat{\beta}) &= \kappa \operatorname*{argmin}_{\Theta,\beta} \|Y\Theta - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_2\|\beta\|_2^2 && (5) \\ & \quad \text{subject to } n^{-1}\|Y\Theta\|_2^2 = 1, && (6) \end{aligned}$$

where Θ is a matrix that yields the optimal scores when applied to indicator matrix Y, and where we add the constraint (7) to avoid degenerate solutions (Grosenick et al. 2008). Given that this is just a sparse version of PDA (Hastie et al. 1995) we have call this combination Sparse Penalized Discriminant Analysis (SPDA) (it has also been called Sparse Discriminant Analysis (SDA) (Clemmensen et al. In Press)). For an interesting alternative approach for constructing sparse linear discriminant classifiers see Witten and Tibshirani (In Press).

2.1.4. Robust loss functions

More generally, we can formulate the penalized regression problem we are interested in as minimizing the penalized empirical risk $\mathcal{R}(\beta)$ as a function of the coefficients, so that

$$\widehat{\beta} = \operatorname*{argmin}_{\beta} \mathcal{R}(\beta) = \operatorname*{argmin}_{\beta} \mathcal{L}(y, \widehat{y}) + \lambda\mathcal{P}(\beta), \tag{7}$$

where $\widehat{y}$ is our estimate of response variable y (note $\widehat{y} = X\widehat{\beta}$ in the linear models we consider) and $\mathcal{L}(y, \widehat{y})$ is a loss function that penalizes differences between the estimated and true values of y. For example, in the above models (3)-(5) we used $\mathcal{L}(y, \widehat{y}) = \|y - \widehat{y}\|_2^2 = \sum_{i=1}^n (y_i - \widehat{y}_i)^2$ ("squared error loss"). While squared error loss enjoys many desirable properties under the assumption of Gaussian noise, it is sensitive to the presence of outliers.

Outlying data points are an important consideration when constructing models for fMRI data, in which a variety of factors from residual motion artifacts to field inhomogeneities can cause some observations to fall far from the sample mean. In the case of standard squared-error loss (as in equations (2)-(5)), these outliers can have undue influence on the model due to the quadratically increasing penalty on the residuals (see Figure 2a). A standard solution in such cases is to use a robust loss function, such as the Huber loss (Huber 2009)

$$\begin{aligned} \mathcal{L}_H(y, \widehat{y}; \delta) &= \sum_{i=1}^N \rho_\delta(y_i - \widehat{y}_i) \qquad (8) \\ \text{where} \qquad & \rho_\delta(y_i - \widehat{y}_i) = \begin{cases} (y_i - \widehat{y}_i)^2/2 & \text{for } |y_i - \widehat{y}_i| \le \delta \\ \delta|y_i - \widehat{y}_i| - \delta^2/2 & \text{for } |y_i - \widehat{y}_i| > \delta \end{cases}. \end{aligned}$$

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

2.2. *Graph-constrained Elastic Net (GraphNet) regression methods and a family of SPDA-based GraphNet classifiers*

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 fit sparse models that do not exclude correlated variables (Zou and Hastie 2005), and that we can turn these regression models into classifiers that perform well when fit to VOI data (Grosenick et al. 2008). However, in a sense the elastic net penalty merely makes the model fitting procedure "blind" to the correlations between input features (by shrinking the sample estimate of the covariance matrix towards the identity matrix). Indeed, if we let λ_2 in equation (5) get very large, this model is equivalent to applying a threshold to mass-univariate OLS regression coefficients (i.e. our estimate of the covariance matrix is a scaled identity matrix) (Zou and Hastie 2005).

In this section, we describe a modification of the elastic net that explicitly imposes structure on the model coefficients. This allows the analyst to pre-specify constraints on the model coefficients based on prior information, physiological constraints, or desirable model properties, and then tune how strongly the model must adhere to these constraints. As the user-specified constraints take the general form of an undirected graph, we have called this regression method the graph-constrained elastic net (GraphNet) (Grosenick et al. 2009; 2010).

The GraphNet model closely resembles the elastic net model, but with a modification to the ℓ_2-norm penalty term:

$$\begin{aligned} \widehat{\beta} &= \kappa \operatorname*{argmin}_{\beta} \|y - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_G\|\beta\|_G^2 \qquad (9) \\ \text{where} \qquad & \|\beta\|_G^2 = \beta^T G \beta = \sum_{j=1}^p \sum_{k=1}^p \beta_j G_{jk} \beta_k, \end{aligned}$$

where G is a sparse graph. Note that in the case where $G = I$, where I denotes the identity matrix, the GraphNet reduces back to the elastic net. Thus the elastic net is a special case of GraphNet and we can replicate the effects of increasing an elastic net penalty by adding a scaled version of the identity matrix $(\lambda_2/\lambda_G)I$ to G (for $\lambda_G > 0$).

2.2.1. GraphNet with the graph Laplacian penalty

The example we will use for the matrix G in the remainder of this paper is the graph Laplacian. If we take the coefficients β to be functions over the brain volume $V \in \mathbb{R}^3$ over time points $T \in \mathbb{R}$ such that $\beta(x, y, z, t)$, then we would like a penalty which penalizes roughness in the coefficients as measured by their derivatives over space and time, such as

$$\mathcal{P}(\beta) = \int_{V,T} \left[\frac{d\beta}{dx} + \frac{d\beta}{dy} + \frac{d\beta}{dz} + \frac{d\beta}{dt}\right]^2 dx\ dy\ dz\ dt = \int_{V,T} \beta^T\ \Delta\beta\ dx\ dy\ dz\ dt. \tag{10}$$

Since we are sampling discretely, we use the discrete version of the Laplacian Δ, the matrix Laplacian L, as described previously (Hastie et al. 1995). This formulation generalizes well to arbitrary graph connectivity and is widely used in spectral clustering techniques (Belkin and Niyogi 2008).

In the case where G is the matrix Laplacian L, the graph penalty, $\|\beta\|_G^2$, has the simple representation

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

where A is the set of index pairs for "adjacent" voxels (and where adjacency is defined by entries in L). When written in this way it is very easy to see how the graph penalty induces smoothness: it penalizes the size of the pairwise differences between adjacent coefficients. If the quadratic terms $(\beta_i - \beta_j)^2$ were replaced by absolute deviations, $|\beta_i - \beta_j|$, then the model would be an instance of the "fused LASSO" (Tibshirani et al. 2005). There are two main reasons we might prefer the use of a quadratic penalty in the present application:

1. The fused LASSO is 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 smoothness often expected in fMRI data.
2. There are significant algorithmic complications that can be avoided by choosing a differentiable penalty on the pairwise differences (Tseng 2001; Friedman et al. 2007a), speeding up model fitting considerably.

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

$$(\widehat{\Theta}, \widehat{\beta}) \quad = \quad \kappa \operatorname*{argmin}_{\Theta,\beta} \|Y\Theta - X\beta\|_2^2 + \lambda_1\|\beta\|_1 + \lambda_G\|\beta\|_G^2 \tag{11}$$

$$\text{subject to } n^{-1}\|\mathrm{Y}\Theta\|_2^2{=}1. \tag{12}$$

We note that in a multi-class classification problem, we would likely want to iteratively minimize over Θ and β (Clemmensen et al. In Press).

2.2.2. Robust GraphNet

Having developed GraphNet using squared-error loss, we can now modify it to include a robust penalty like the Huber loss defined above. Replacing the squared error loss function with the loss function (8) we have

$$\widehat{\beta} = \kappa \operatorname*{argmin}_{\beta} \mathcal{L}_H(y, X\beta; \delta) + \lambda_1\|\beta\|_1 + \lambda_G\|\beta\|_G^2. \tag{13}$$

We define the SPDA-RGN classifier similarly to that for the standard GraphNet (11). Note, however, that the model now has an addition hyperparameter to either be fit (or assumed): the value of δ that determines where the loss function switches from the quadratic to the linear regime (Figure 2a). Further, the loss function on the residuals is no longer quadratic and therefore could lead to much slower convergence of our optimization method. We discuss a solution to this issue next.

2.2.3. Infimal convolution for non-quadratic loss functions

Convergence speed of subgradient methods such as coordinate-wise descent is greatly improved if the loss function has a quadradic form. Non-quadratic loss functions like the Huber and the Huberized-hinge loss functions are not quadradic as written in 8, and thus could take numerous iterations to converage for each coefficient—significantly increasing computation time. Indeed, these methods are not amenable to optimization using coordinate descent in general, as their objectives are written as conditional functions with different loss conditions parameterized by δ. However, we can circumvent these problems and extend the applicability of coordinate-wise descent methonds using a trick from convex analysis to rewrite these loss functions as quadradic forms in an augmented set of variables. This method, called infimal convolution (Rockafellar 1970), is defined as

$$(f \star_{\text{inf}} g)(x) := \inf_y \{f(x-y) + g(y) | y \in \mathbb{R}^n\}, \tag{14}$$

where f and g are two functions of $x \in \mathbb{R}^p$. In this way it is possible to rewrite the ith term in the the Huber loss function (8) as the infimal convolution of the squared and absolute-value functions applied to the ith residual r_i:

$$\rho_\delta(r_i) = ((1/2)(\cdot)^2 \star_{\text{inf}} |\cdot|)(r_i) = \inf_{a_i+b_i=r_i} a_i^2/2 + \delta|b_i|, \tag{15}$$

where $r_i = y_i - (X\widehat{\beta})_i$. This yields the augmented estimation problem

$$\widehat{\alpha}, \widehat{\beta} = \underset{\alpha,\beta}{\operatorname{argmin}} \; (1/2)\|y - X\beta - \alpha\|_2^2 + \lambda_G \beta^T G \beta + \delta\|\alpha\|_1 + \lambda_1 \|\beta\|_1, \tag{16}$$

which we can rewrite as

$$\begin{aligned} \widehat{\gamma} &= \underset{\gamma}{\operatorname{argmin}} \; (1/2)\|y - Z\gamma\|_2^2 + \lambda_G \gamma^T G' \gamma + \sum_{j=1}^{p+n} w_j |\gamma_j| \\ & Z = [X \quad I_{n\times n}], \quad \gamma = [\beta \quad \alpha], \quad w_j = \begin{cases} \lambda_1 & j = 1, .., p \\ \delta & j = p+n \end{cases}, \\ & G' = \begin{bmatrix} G & 0_{1\times n} \\ 0_{n\times 1} & 0_{n\times n} \end{bmatrix} \in S_+^{(p+n)\times(p+n)}, \end{aligned} \tag{17}$$

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 section 2.4 below. After solving for augmented coefficients $\widehat{\gamma}$ we can simply discard the last n coefficients to yield $\widehat{\beta}$. A similar approach can be taken with the hinge-loss of a support vector machine classifier (as we show next), or more generally with any loss function decomposable into an infimal convolution of convex functions (see Appendix).

2.2.4. Support Vector Machine (SVM) GraphNet for classification

In the $n << p$ classification problem, maximum-margin classifiers such as the support vector machine often perform exceedingly well in terms of classification accuracy, but do not yield readily interpretable coefficients. For this reason we also develop a sparse SVM with graph constraints (SVM-GraphNet) related to the "Hybrid Huberized SVM" of (Wang et al. 2008) as an alternative to the SPDA method. Using a 'Huberized-hinge' loss function on the model residuals we have

$$\widehat{\beta} = \kappa \operatorname*{argmin}_{\beta} \mathcal{L}_{HH}(y^T X\beta; \delta) + \lambda_1 \|\beta\|_1 + \lambda_G \|\beta\|_G^2, \tag{18}$$

where $y \in \{-1, 1\}$. Letting $\widehat{y} = X\widehat{\beta}$ be our estimates of the target variable,

$$\mathcal{L}_{HH}(y, \widehat{y}; \delta) = \sum_{i=1}^{N} \rho_\delta(y_i, \widehat{y}_i) \tag{19}$$

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

which is the Huberized-hinge loss of (Wang et al. 2008). As with the Huber loss there is an additional hyperparameter δ to be fit or assumed. In this case δ determines where the hinge-loss function switches from the quadratic to the linear regime (see Figure 2a). This problem's loss function can also be written using infimal convolution to yield a more convenient quadratic objective term (see Appendix).

2.2.5. Adaptive GraphNet models

Automatic variable selection using the methods above is based on an approach that shrinks coefficient estimates towards zero, resulting in downwardly biased coefficient magnitudes. This (a) makes it difficult to interpret coefficient magnitude in terms of original data units, and (b) severely restricts the conditions under which the lasso can perform consistent variable selection (Zou 2006). Ideally our model would select the correct parsimonious set of features (the "true model", were it known), but avoid shrinking the nonzero coefficients that are kept in the model (unbiased estimation)—desiderata known as the "oracle" property (Fan and Li 2001).

Several estimators possessing the oracle property (given certain conditions on the data) have been reported in the literature, including the adaptive lasso (Zou 2006; Zhou et al. 2011) and the adaptive elastic net (Zou and Zhang 2009). These estimators are simple modifications of penalized linear models. They work by starting with some initial estimates of the coefficients and use these to adaptively reweight the penalty on each individual coefficient β_j. We rewrite the robust GraphNet to have an adaptive penalty (the robust adaptive GraphNet) as follows

$$\widehat{\beta}^{RAGN} = \operatorname*{argmin}_{\beta} \mathcal{L}_H(y, X\beta; \delta) + \lambda_1 \sum_{j=1}^{p} \widehat{w}_j |\beta_j| + \lambda_G ||\beta||_G^2 \tag{20}$$

$$\widehat{w}_j = \left|\tilde{\beta}_j\right|^{-\gamma}. \tag{21}$$

The idea here is that important coefficients will have large starting estimates $\tilde{\beta}_j$ (where $\tilde{\beta}_j$ is an unbiased estimator of β_j) and therefore they will be shrunk at a rate inversely proportional to their unbiased starting estimates, leaving them unbiased in the final model. Coefficients with small starting estimates $\tilde{\beta}_j$ will experience additional shrinking and will likely be excluded from the model. We let $\gamma = 1$ as used previously in the finite sample case (Zou 2006; Zou and Zhang 2009), and by analogy to the adaptive elastic net (Zou and Zhang 2009) use $\tilde{\beta} = \hat{\beta}^{RGN}$ for some fixed values of λ_G and δ (chosen based on the robust GraphNet performance at those values).

It is important to note that the oracle properties that hold in the asymptotic case may not apply to our finite sample, $p >> n$ situation. Nevertheless we include these models for comparison as such oracle properties are desirable and there is evidence suggesting that because the adaptive elastic-net deals well with collinearity arising from having many correlated inputs, it has improved finite sample performance for $n >> p$ problems (Zou and Zhang 2009).

Finally, we discuss a heuristic alternative to adaptive methods for adjusting nonzero coefficient magnitudes to be on the scale of the original data. This approach can be used with any of the above models.

2.2.6. Coefficient rescaling

The elastic net was originally formulated by (Zou and Hastie 2005) in both a "naive" and a rescaled form. The authors note that a combination of an ℓ_1 and ℓ_2 penalty can "double shrink" the coefficients. To correct this they propose rescaling the "naive" solution by a factor of $\kappa = 1 + \lambda_2$ (Zou and Hastie 2005). Heuristically, the aim is to retain the desirable variable selection properties of the elastic net while rescaling the coefficients to be closer to the original scale. However, it is not clear that $\kappa = 1 + \lambda_2$ is the correct multiplicative factor if the data are collinear, and this can complicate the problem of choosing a final set of coefficients for a model.

A simple alternative is to fit the elastic net, generating a fitted response $\hat{y}$, and then to regress y on $\hat{y}$. In particular, solving the simple linear regression problem

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

yields an estimate $\widehat{\kappa}$ that can be used to rescale the coefficients obtained from fitting the Elastic Net (Daniela Witten and Robert Tibshirani, personal communication). The intuitive motivation is that this should produce the $\widehat{\kappa}$ which puts $\widehat{\beta}$ and $\widehat{y}$ on a reasonable scale for modeling y.

Besides its simplicity, the principal advantage of this approach is that is requires no analytical knowledge about the amount of shrinkage that occurs as λ_2 is increased. This is particularly appealing because the same strategy of regressing $\hat{y}$ on y can be used with more general problems, such as the GraphNet. For example, the "double shrinking" caused by the graph penalty can be corrected in this manner.

Finally, we describe one important exception in the binary classification case with balanced data that has been centered. If we are fitting a linear model of the form $Y = X\beta$, but we have recentered the data such that we fit

$$Y - \bar{Y} = \sum_j (X_j - \bar{X}_j)\beta_j,$$

then the model for Y is

$$Y = \sum_j X_j\beta_j + \bar{Y} - \sum_j \bar{X}_j\beta_j,$$

and the intercept we are implicity fitting is given by $-\sum_j \bar{X}_j\beta_j$. The balancing of trials ensures that $\bar{Y}$ is always zero. This is very convenient, for if we consider a new model $\tilde{\beta} = \kappa\beta$ we don't need to refit the intercept, and the predictions become

$$\widehat{Y}' = \kappa\left(\sum_j X_j\beta_j - \sum \bar{X}_j\beta_j\right) = \kappa\widehat{Y}.$$

This means that the balanced two-class classification problem on centered data does not depend on any scaling of parameters, even with the intercept term in the model.

2.3. Interpreting GraphNet regression and discriminant models

2.3.1. GraphNet as as constrained maximum likelihood problem

Given some observed fMRI data, we would like to maximize the likelihood of our model given the data, subject to some constraints on the model coefficients–specifically that they are sparse and structured. One way of writing this goal more formally is

$$\begin{aligned} \underset{\beta}{\text{maximize}} \quad & \text{loglik}(\beta|X, y) && (22) \\ \text{subject to} \quad & \|\beta\|_1 \leq c_1 && (23) \\ & \|\beta\|_G^2 \leq c_G. && (24) \end{aligned}$$

The particular log-likelihood function in (22) is determined by the distribution that we assume for the observation noise in equation (1), and the constraints $c_1 \in \mathbb{R}_+$ and $c_G \in \mathbb{R}_+$ are hard bounds on the size of the coefficients in the ℓ_1 and ℓ_G norms, respectively. Assuming identical and independently distributed (i.i.d.) additive Gaussian noise results in the squared-error loss function used in the standard GraphNet problem (9), while a mixture of Gaussian and Laplacian noise yields the Huber loss (8), with the particular mixture proportions set by the parameter δ (Huber 2009).

If for desired values of the hard constraints c_1 and c_G are somehow known beforehand, a globally optimal solution to (22)-(24) can be found using a projected subgradient method on the dual problem. In fMRI analysis, however, a priori values for c_1 and c_G are rarely known in advance. Therefore, instead of solving the problem for fixed for c_1 and c_G, we consider the Lagrangian relaxation,

$$L(\beta, \lambda_1, \lambda_G) = -\text{loglik}(\beta|X, y) + \lambda_1(\|\beta\|_1 - c_1) + \lambda_G(\|\beta\|_G^2 - c_G), \ \lambda_1, \lambda_G \in \mathbb{R}_+, \qquad (25)$$

where the hard constraints c_1 and c_G have been replaced by nonnegative linear penalties λ_1 and λ_G (Boyd and Vandenberghe 2004). When this function is minimized with respect to β, the c_1 and c_G terms disappear, so given particular values of λ_1 and λ_G, we can equivalently solve the penalized maximum likelihood problem

$$\widehat{\beta} = \underset{\beta}{\text{argmin}} \ -\text{loglik}(\beta|X, y) + \lambda_1\|\beta\|_1 + \lambda_G\|\beta\|_G^2, \ \lambda_1, \lambda_G \in \mathbb{R}_+, \qquad (26)$$

characteristic of the GraphNet estimators—with the particular log-likelihood function determining which loss function $\mathcal{L}$ is used. This formulation leads to the following interpretations of the model parameters and model coefficients.

2.3.2. Interpreting model parameters: dual variables as prices

In the Lagrangian formulation, the 'dual variables' λ_1 and λ_G express our (linear) irritation with any violation of the constraints. Since we solve problem 26, c_1 and c_G are effectively zero, and we are penalized for any deviation of the coefficients from zero. We can thus interpret λ_1 and λ_G as the prices we are willing to pay to improve our model likelihood at the expense of a less sparse or less structured solution, respectively.

Examining the sensitivity of model fit to different values of λ_1 and λ_G can therefore tell us about underlying structure in the data. For example, if the neural activity related to a particular cognitive task was very sparse and highly localized in a few uncorrelated voxels, then we should be willing to pay more for sparsity and less for smoothness (large λ_1, small λ_G). In contrast, if large smooth and correlated regions underlie the task then tolerating a large λ_G could help our model fit substantially. To explore such possibilities, we can plot median test rates from cross validations at different combinations of model parameters. Figure 5 shows example contour plots of median rates as as function of λ_1 and λ_G over the parameter grid on which we fit the GraphNet and robust GraphNet classifiers.

2.3.3. Interpreting model coefficients

Problem 26 can also be arrived at from a Bayesian perspective as a maximum a posteriori probability (MAP) estimator. In this case, the form of the penalty $\mathcal{P}(\beta)$ is related to our prior beliefs about the structure of the model coefficients. For example, under the well-known equivalence of penalized regression techniques and posterior modes, the elastic net penalty corresponds to the prior

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

(Zou and Hastie 2005). The GraphNet penalty thus corresponds to the prior

$$\begin{aligned} p_{\lambda_1,\lambda_G}(\beta) &\propto \exp\left\{-\left(\lambda_1||\beta||_1 + \lambda_G\beta^T G\beta\right)\right\} \\ &\propto \prod_{i=1}^{p}\exp\left\{-\lambda_1|\beta_j|\right\}\prod_{i=1}^{p}\exp\left\{-\lambda_G\sum_{i\sim j}\beta_i G_{ij}\beta_j\right\}, \end{aligned} \tag{27}$$

where $i \sim j$ denotes that node i in the graph G is adjacent to node j. Therefore, the GraphNet problems are also equivalent to a MAP estimator of the coefficients with a prior consisting of a convex combination of a global Laplacian (double-exponential) and a local Markov Random Field (MRF) prior. In other words, they explicitly take into account our prior beliefs about the model coefficients being globally sparse but locally structured.

It is important to note that because this is a binary classification task, the bias-variance relationship is more complicated than that found in the textbook squared-error loss case (where there is a simple additive relationship). In particular, due to a multiplicative interaction between bias and variance in the case of 0-1 loss, models with negative decision boundary bias are insensitive to "over-smoothing" in the balanced sample case Friedman (1997). This is in contrast to regression or class probability estimation where "over-smoothing" degrades performance much more sharply.

As discussed above, OS can be used to turn regression methods into discriminant classifiers. This allows us to use notions from regression like degrees of freedom, as well as those utilized in discriminant analysis, such as class visualization in the discriminant space using discriminant coordinates, or trial-by-trial posterior probabilities for individual observations (Hastie et al. 1995). Having trial-by-trial probabilities of observations being in one class or another may prove particularly interesting the in functional imaging setting, where behavioral variables are available at the single trial level (e.g. reaction time, product preference).

2.4. Model fitting and computational considerations

2.4.1. Coordinate-wise decent and active set methods

Fitting regression models to whole-brain fMRI data demands efficient computational methods. This is particularly true if we would like to cross-validate the model fit over a grid of possible parameter values. In the application below (section 2.5), we have 26,630 input features (voxels) at each of 7 time points. Fitting the adaptive robust GraphNet using leave-one-subject out (LOSO) cross-validation (25 fits) for each realization of possible parameter values over a $90 \times 5 \times 6 \times 10 \times 3$ grid of possible values $\{\lambda_1, G, \lambda_G, \delta, \lambda_1^*\}$ requires $2,025,000$ model fits on $1,882$ observations of $186,410$ input features.

To make millions of model fits over hundreds of thousands of input features feasible, we formulated our minimization problems—equations (9)(13) and (18)—as coordinate-wise optimization problems (Tseng 2001) solved using active set methods (Friedman and Tibshirani 2010). In this formulation we fit one coefficient value at a time ("coordinate-wise" descent), holding the rest constant, and keep an "active set" of those nonzero coefficients that improve the fit enough to not being shrunk out of the model (to zero). We start with a large value of λ_1, corresponding to all coefficients being zero, and then slowly decreasing λ_1 to allow more and more coefficients into the model. This allows us to consider an 'active set' of the model coefficients rather than all 186,410 inputs at each coordinate-wise update. Occasional sweeps though all the coefficients are made to look for new variables to include, as in (Friedman and Tibshirani 2010). We stop decreasing λ_1 before it reaches zero, as this would yield a dense solution that would both be computationally expensive and yield poor estimates (Hastie et al. 2009).

In using coordinate-wise descent, we rely on the fact that our regression methods can be formulated as

$$\underset{\beta}{\operatorname{argmin}}\ f(\beta_1, ..., \beta_p) = \underset{\beta}{\operatorname{argmin}}\ g(\beta_1, .., \beta_p) + \sum_{j=1}^{p} h(\beta_j) \tag{28}$$

where $g(\beta_1, .., \beta_p)$ is a convex, differentiable function (e.g., squared-error and Huber loss plus our quadratic penalty $||\beta||_G^2$), and where each $h(\beta_j)$ is a convex (but not necessarily differentiable) function (e.g., the ℓ_1 penalty). If the convex, non-differentiable part of our penalty is separable in coordinates β_j—as is true of $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$ —then coordinate descent can be shown to converge to global solution of the minimization problem (Tseng 2001). In the case of Huber loss or Huberized-hinge loss, we first rewrite the two-part loss function as a single quadratic loss function using infimal convolution as described above.

As a specific example of solving a GraphNet problem with coordinate descent, consider the coordinate-wise updates for the standard GraphNet problem given in equation (9). Letting $\hat{y} = \tilde{X}\tilde{\beta} + X_{\cdot j}\beta_j$ (where $\tilde{X} = X_{\cdot \neq j}$ is the matrix X with the jth column removed, and $\tilde{\beta} = \beta_{\neq j}$ the coefficient vector with the jth coefficient removed), the gradient of the risk with respect to just the jth coefficient (holding the others fixed) is

$$\nabla_{\beta_j}\mathcal{R} = -X_{\cdot j}^T y + X_{\cdot j}^T \tilde{X}\tilde{\beta} + X_{\cdot j}^T X_{\cdot j}\beta_j + (\lambda_2/2)\tilde{\beta}^T (G_{\neq j}\cdot)_{\cdot j} + \lambda_2 G_{jj}\beta_j + (\lambda_1/2)\Gamma(\beta_j) \tag{29}$$

where the subgradient $\Gamma(\beta_j) = -1$ if $\beta_j < 0$, $\Gamma(\beta_j) = 1$ if $\beta_j > 0$ and $\Gamma(\beta_j) \in [-1, 1]$ if $\beta_j = 0$. Basic algebra then gives the coordinate update iterate for the jth coefficient estimate:

$$\hat{\beta}_j \leftarrow \frac{S\left(X_{\cdot j}^T(y - \tilde{X}\tilde{\beta}) - (\lambda_2/2)\tilde{\beta}^T(G_{\neq j}\cdot)_{\cdot j},\ \lambda_1/2\right)}{X_{\cdot j}^T X_{\cdot j} + \lambda_2 G_{jj}} \tag{30}$$

where $S(x, \gamma) = \text{sign}(x)(|x| - \gamma)_+$ is the soft-thresholding function (Friedman et al. 2007a). Note that if graph $G = I$, and the columns of X are standardized to have unit norm we recover the coordinate-wise Elastic Net update der Kooij (2007); Friedman et al. (2007b).

2.4.2. Computational complexity

Looking more closely at equation 31, we see that if the variables are standardized (such that $X_{\cdot j}^T X_{\cdot j} = 1$) then the $(c+1)$st coefficient update for the jth coordinate can be rewritten

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

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 (31) until the new estimates converge, at which point λ_1 is decreased incrementally and another $O(pn)$ sweep 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 model fit stops improving or a pre-specified coefficient density is reached. Let $a = \sum_{i=1}^{l} a_i$ denote the total number coefficients 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. 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 than $m = p^2 - p$ and the $O(lm)$ term will dominate.

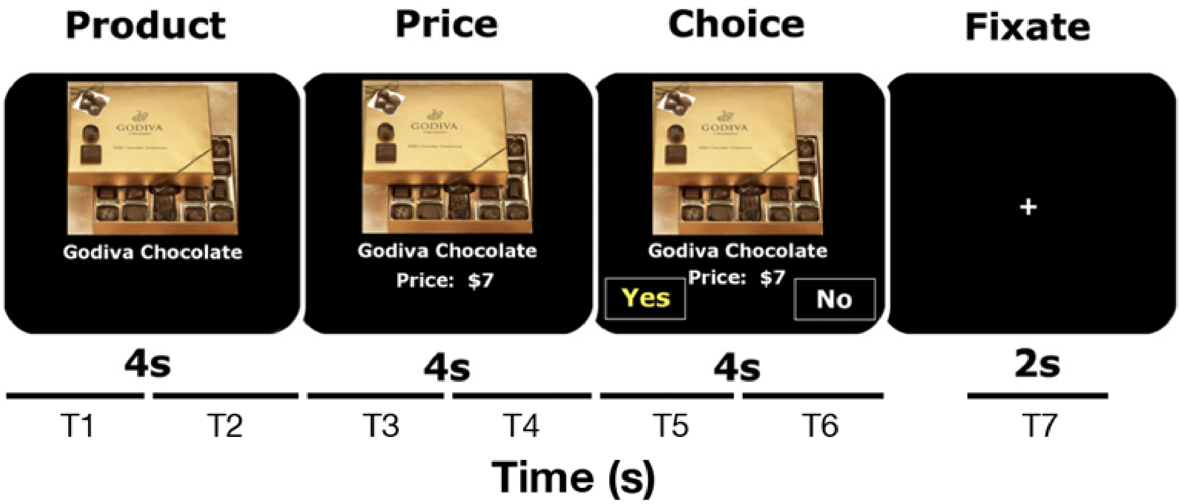

Figure 3. Save Holdings, or Purchase (SHOP) task trial structure. Subjects saw a labeled product (product period; 4 s, 2 TRs), saw the product's price (price period; 4 s, 2TRs), and then chose either to purchase the product or not (by selecting either ''yes'' or ''no'' presented randomly on the right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

2.4.3. Cross validation, classification accuracy, and parameter tuning

For classification, data were down-sampled to ensure equal numbers of trials per class. This allowed us to conservatively assess significance of binary classification rates using the binomial distribution with a predicted 50% success rate. For 10-fold cross validation, this down-sampling occurred once so that the overall dataset was balanced, and then observations were assigned to folds. In the leave-one-subject-out cross-validation the down-sampling was done on a per-subject basis, so that it would be within folds (each subject's data is a test fold). We remark that the range for these grid values we chosen based off some preliminary fits, and that more efficient adaptive and analytical approaches to parameter search may be more effective. The grid values used are given in the Appendix.

2.5. Application: Predicting buying behavior using fMRI (Knutson et al., 2007)

2.5.1. Subjects and SHOP task

Data from 25 healthy right-handed subjects were included in these analyses (one subject's original fMRI data could not be recovered and so was omitted). Along with the typical magnetic resonance exclusions (e.g., metal in the body), subjects were screened for psychotropic drugs and 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. In addition to the 25 subjects who were included in the analysis, 6 subjects who purchased fewer than four items per session (i.e., $<$ 10%) were excluded due to insufficient data to model, and 8 subjects who moved excessive amounts (i.e., $>$ 2 mm between whole brain acquisitions) were excluded.

While being scanned, subjects participated in a "Save Holdings Or Purchase" (SHOP) Task (Figure 3). During each task trial, subjects saw a labeled product (product period; 4 sec), saw the product's price (price period; 4 sec), 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 sec), before fixating on a crosshair (2 sec) prior to the onset of the next trial (see Supplement 1 for illustration of the task layout).

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-$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 the products 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 model.

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 in the same pseudo random order (item sets were counterbalanced across subjects). We consider only data from the first time the item was presented here (see (Grosenick et al. 2008) for a comparison between first and second 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.

2.5.2. 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 X 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).

2.5.3. 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. 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, and result in rougher coefficients overall (Hastie et al. 1995).

Spatiotemporal data were arranged as in previous spatiotemporal analyses (Mourao-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 seconds), yielding a total of $p = 186,410$ input input features per trial. Altogether, these data included $n = 1,882$ trials across the 25 subjects.

3. Results

3.1. Classification rates

Table 1: Median accuracy and parameters for SPDA and SVM classifiers fit with Leave-5-Subjects-Out (L5SO) cross-validation.

Model	Training	Test	Sparse	Correlated	Structured	Robust	Adaptive
Lasso[1]	98.75%	68.5%	$\lambda_1 = 33$				
Elastic Net[2]	90.38%	72.5%	$\lambda_1 = 54$	$\lambda_2 = 10000$			
GraphNet[3] (GN)	86.86%	73.73%	$\lambda_1 = 68$	$\lambda_2 = 1000$	$\lambda_G = 100$		
Robust GN (RGN)	86.75%	**74.5**%	$\lambda_1 = 43$	$\lambda_2 = 100$	$\lambda_G = 100$	$\delta = 0.3$	
RGN + temporal	96.5%	73.75%	$\lambda_1 = 42$	$\lambda_2 = 1000$	$\lambda_G = 10$	$\delta = 0.5$	
Adaptive RGN	91.38%	73.75%	$\lambda_1 = 50$	$\lambda_2 = 10000$	$\lambda_G = 100$	$\delta = 0.4$	$\lambda_1^* = 0.01$
ARGN + temporal	90.75%	73.5%	$\lambda_1 = 40$	$\lambda_2 = 1000$	$\lambda_G = 100$	$\delta = 0.3$	$\lambda_1^* = 0.01$
GraphSVM	85.3%	73.0%	$\lambda_1 = 120$	$\lambda_2 = 1000$	$\lambda_G = 10$	$\delta = 0.5$	
Linear SVM[4]	97.94%	71.0%		$C = 3.8 \times 10^{-6}$		✓	

[1](Tibshirani 1996), [2](Zou and Hastie 2005), [3](Grosenick et al. 2009), [4]Cortes and Vapnik (1995). Chance level is 50%.

If models that use brain data to predict choice generalize well across subjects, it implies an invariance in the neural substrates of choice across individuals. We compared the abilities of the proposed SPDA models to predict eventual choices to purchase with a more commonly used linear SVM model, examining both generalization to held-out test sets from data pooled across subjects, and to test sets consisting of subjects held out entirely from the fitting procedure. Best results for the SPDA classifiers and linear SVM fit across the 25 subjects (and the model parameters for each) are given in Tables 1 and 2, as well as a summary of each model's properties. Models were fit using either leave-one-subject-out (LOSO, Table 1) leave-5-subjects-out (L5SO, Table 2) cross-validation, and we report both training and test results to examine to what extent the models overfit the training data.

3.1.1. Generalization to held-out subjects (L5SO cross-validation)

Best median training and test rates are given for SPDA models fit over the grid of parameters given in (29) and the SVM parameters given in (32). Despite a more than 1000-fold increase in the number of input features relative to previous region of interest analyses (Grosenick et al. 2008), the whole-brain classifiers performed significantly better than previous ROI-based models Knutson et al. (2007); Grosenick et al. (2008). Further, we note that robust models may be more effective on this data, as decreasing δ in the SPDA-RGN (making it more robust, see Figure 2) improved its rates over the GraphNet. This is also consistent with the SVM performing well, as the hinge loss employed by the the SVM is robust against outliers. Still, these effects are less pronounced than the improvement of the GraphNet methods over ℓ_1 penalization alone. Note that the lasso and linear SVM tend to overfit the training data more, while the SPDA classifiers have more modest training rates. Overall, the best median L5SO rate was from the robust GraphNet, with accuracy

Table 2: Median accuracy and parameters for SPDA and SVM classifiers fit with Leave-One-Subject-Out (LOSO) cross-validation.

Model	Training	Test	Sparse	Correlated	Structured	Robust	Adaptive
Lasso[1]	90.52%	68.75%	$\lambda_1 = 66$				
Elastic Net[2]	90.83%	70.0%	$\lambda_1 = 64$	$\lambda_2 = 1000$			
GraphNet[3] (GN)	87.5%	71.25%	$\lambda_1 = 59$	$\lambda_2 = 10000$	$\lambda_G = 1000$		
Robust GN (RGN)	83.75%	72.5%	$\lambda_1 = 26$	$\lambda_2 = 1000$	$\lambda_G = 1000$	$\delta = 0.2$	
RGN + temporal	83.75%	72.5%	$\lambda_1 = 35$	$\lambda_2 = 1000$	$\lambda_G = 1000$	$\delta = 0.3$	
Adaptive RGN (ARGN)	85.4%	72.5%	$\lambda_1 = 20$	$\lambda_2 = 1000$	$\lambda_G = 1000$	$\delta = 0.2$	$\lambda_1^* = 0.01$
ARGN + temporal	88.33%	**73.75%**	$\lambda_1 = 30$	$\lambda_2 = 1000$	$\lambda_G = 100$	$\delta = 0.2$	$\lambda_1^* = 0.01$
GraphSVM	89.48%	**73.75%**	$\lambda_1 = 88$	$\lambda_2 = 100$	$\lambda_G = 100$	$\delta = 0.5$	
Linear SVM[4]	91.56%	68.75%		$C = 7.6 \times 10^{-6}$		✓	

[1](Tibshirani 1996), [2](Zou and Hastie 2005), [3](Grosenick et al. 2009), [4]Cortes and Vapnik (1995). Chance level is 50%.

on held-out test data of 74.5% (the linear SVM accuracy was 71.0%). Examing the distribution of the rates across the 25 folds, Figure 4b shows that the SVM appears to have less variance across fits to held-out subjects. The marked non-normality of these distributions is interesting, and motivated our reporting the median rather than mean accuracy over cross-validation folds.

3.1.2. Generalization to held-out individuals (LOSO cross-validation)

In addition to cross-validation over pooled data across subjects, we also ran leave-one-subject-out (LOSO) cross validation (i.e., using the data from 24 subjects to predict results for each remaining subject). Repeating this procedure for all subjects yielded one held-out classification rate per subject, indicating how well the group model generalized to that subject. Repeating this for all subjects yielded one held-out test rate per subject. This rate indicates how well the model fit to all the other subjects generalized to the held-out subject—a measure that an approach that may be of interest in studies of individual differences. Figure 4a shows smoothed histograms of the LOSO classification rates for the SPDA-RGN and SVM analyses. Overall, the GraphNet methods outperform the linear SVM on LOSO cross-validation. Critically, not only are the GraphNet methods more performant, but they are also interpretable.

3.2. Visualization and interpretation of coefficients and parameters

3.2.1. Interpreting model coefficients

Although SPDA-RGN and SVM models classified future purchase choices similarly overall, SPDA models have the comparative advantage of producing interpretable results. Given the temporal as well as spatial res-

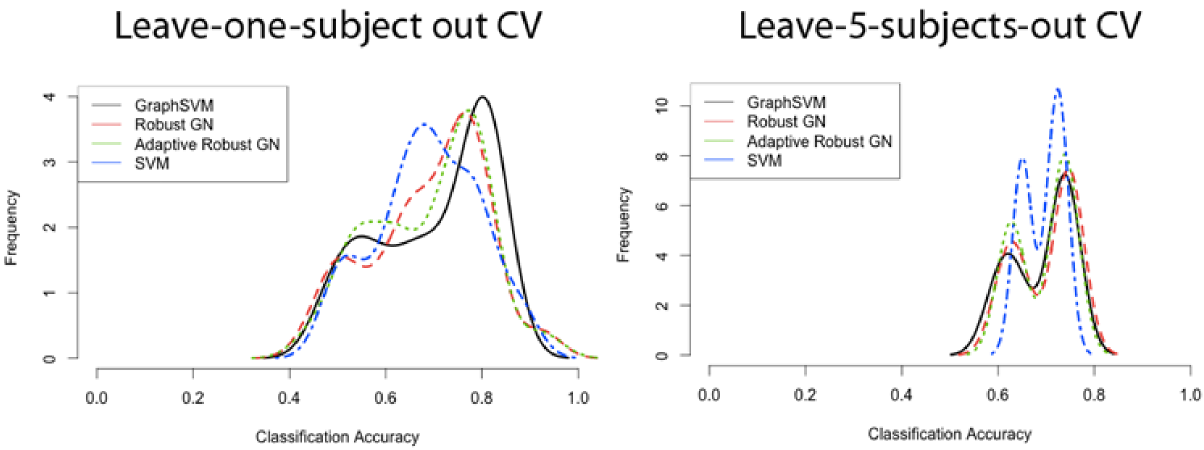

Figure 4. Smoothed histogram densities of leave-one-subject out test rates. Models were fit to all subjects except one, and then tested on the held-out subject. This was done for all subjects and smoothed histograms of these rates were calculated for the Robust GraphNet ($\delta = 1$, blue lines) and linear SVM ($C = 4$, red lines).

olution of the present dataset, interpretation might extend not only to which brain regions predict purchasing choices, but also when. Thus, an investigator who knows when different events occurred (and accounts for the lag in the hemodynamic response) might infer that different events promoted eventual purchasing choices by altering activation in specific regions. Applied to the SHOP dataset, SPDA-RGN revealed several novel findings (Figure 5). First, consistent with previous reports (Knutson et al. 2007; Grosenick et al. 2008), nucleus accumbens (NAcc) activation began to positively predict purchase choices at the time of product presentation, and this persisted throughout the price period. Medial prefrontal cortex (MPFC) and midbrain activation, on the other hand, began to positively predict purchase choices when the price appeared (but not during the product presentation period). Additionally, and not included in previous findings, posterior cingulate activation also began to positively predict purchase choices during price presentation. Reassuringly, no regions predicted purchase choices during fixation presentation. Finally, we note that the best models chooses far more voxels that positively than negatively predict purchasing.

3.2.2. Interpreting model parameters

Figure 6 shows contour plots of median L5SO cross-validation rates over the values $\{\lambda_1, \lambda_G, G\}$ (29) on which we fit the GraphNet and robust GraphNet (results shown are for $\delta = 0.3$).Overall we find that for this particular data set that there is an optimal range of parameters that deliver similar model performance and coefficient values. Sparser models such as the lasso do not perform well, as shown by the dotted white line in Figure 6a. Comparing the rates in Figure 6 suggests that for this data a certain amount of coefficient smoothness and inclusion of correlated variables in the final model is important.

4. Discussion and Conclusions

4.1. Application to SHOP task data

With respect to the specific application to prediction of choice, the current methods offer several advantages over previous region of interest based analyses. Reassuringly, the GraphNet classifiers delivered findings that overlapped with prior region of interest based results, replicating the observation that nucleus accumbens (NAcc) activation begins to predict purchase choices during product presentation while medial prefrontal cortical (MPFC) activation begins to predict purchase choices during price presentation. It is also interesting to note areas that were not included by the models. While one account posits that in the context of fMRI, NAcc activation indexes gain predictions (Knutson et al. 2001; Knutson and Greer 2008), an alternative account posits that NAcc activation instead indexes gain prediction errors, while MPFC activation indexes gain predictions (e.g., Hare et al. 2008). To the extent that gain predictions forecast future events while gain prediction errors reflect past events, the gain prediction account posits that NAcc activation in response to products should predict later purchase choices, whereas the gain prediction error account instead posits that MPFC activation in response to products should predict later purchase choices. SPDA-RGN results clearly support the gain prediction account, since NAcc activation in response to products predict choices to purchase, whereas MPFC activation does not (instead predicting choice in response to later presented price information – consistent with a value integration account). SPDA-RGN also revealed a previously unnoticed finding in which posterior cingulate activation clearly appears to predict purchase choices at price presentation. Accounts of posterior cingulate function in valuation remain less developed than similar accounts of NAcc and MPFC function. Nonetheless, this result might be consistent with attentional and salience-based accounts of posterior cingulate function McCoy et al. (2003) and highlights a region deserving of future study in the context of predicting choice.

4.2. Interpretable models for whole-brain fMRI

We set out to design and develop a model for fMRI data that was able to yield interpretable results for whole-brain data over multiple time-points while yielding accuracy competitive with current state-of-the-art methods for large data. In addition we hoped that this model would choose relevant features in a principled way–including all relevant features while excluding nuisance parameters–and allow us to flexibly constrain model coefficients using prior information. Further, we hoped the resulting coefficients would be interpretable in the native data space (voxel time series) and that their magnitude would be relatively unbiased despite the shrinkage methods employed to yield sparsity.

The GraphNet-based models presented here are a good first pass at providing these desirable–and at times competing–model properties. In particular, the adaptive robust GraphNet allows automatic variable selection (and is an adaptation of models that are asymptotically model selection consistent Zou and Zhang (2009)), incorporation of prior information in the form of a graph penalty, and yields minimally biased coefficients as a result of adaptive reweighting. These methods can be used in either the regression or classification setting (using optimal scoring), and our current application suggests competitive rates with state-of-the-art classifiers for high dimensional data.

Still, this is only one example application, and further validation of performance is necessary. In addition, several caveats should be considered. First, the adaptive methods used to "unshrink" model coefficients may be unstable in the small sample case, as we discuss above, so heuristic methods like the one described above may be more effective in some circumstances. As the valance and relative magnitude of the per-voxel coefficients may be more important for interpretation than their absolute magnitude in many cases, this may not be as important as variable selection consistency, which dictates which voxels are considered for interpretation. However, the model selection consistency results established for the elastic net (Jia and Yu 2009; De Mol et al. 2009) and adaptive elastic net (Zou and Zhang 2009) are recent and must be further validated and carefully extended to the GraphNet models.

Finally, a critic could justly claim that a factor analysis (or other projection onto a lower dimensional manifold) could be used to reduce the number of features prior to applying regression or classification methods to potentially improve computational speed and results. This would of course depend on the efficiency of their dimensionality reduction method, and would not guarantee results interpretable in terms of the original data space. However, some previous work suggests that factor analytic methods applied to functional data, when combined with a sparse regression method to choose latent factors of interest, can yield satisfactory results for very large data (millions of input features), and can be projected back into the original data space for interpretation (Grosenick et al. In Press; Wager et al. 2011). Our results here suggest that such an approach paired with recently developed methods for sparse, structured PCA (Allen et al. Submitted) would be particularly useful in fMRI applications.

4.3. Future directions

There are a variety of graph constraints besides the spatial matrix Laplacian to explore, including (1) weighted graphs derived from the data to be adaptive to local smoothness, (2) cut-graphs derived from segmented brain atlases that would allow adjacent regions known to be functionally distinct to be penalized independently, and (3) weighted graphs derived from diffusion data, which would allow constraints on voxels adjacent on an connectivity graph rather than in space or time. In addition, using the goodness-of-fit measure provided by the model to make inferences about which of a set of structural graphs best relates to functional data, or to adaptively alter graph weights to explore structure in functional data. Finally, all of the methods considered above model relationships between input features and the target variable as linear. While this approximation works well in many cases, signal saturation effects alone dictate that it be false. Recently, nonlinear methods based on scatterplot smoothers have been developed that work well with coordinate-wise methods (Ravikumar et al. 2009). We intend to explore these along with structured feature selection methods to yield more realistic interpretable models for fMRI data.

Appendix A

Robust GraphNet: coordinate-wise coefficient updates using infimal convolution

For a particular coordinate j , we are interested in the estimates

$$\begin{aligned}
\widehat{\gamma}_j &= \underset{\gamma_j}{\text{argmin}}\ (1/2)\|y - \tilde{Z}\tilde{\gamma} - Z_{\cdot j}\gamma_j\|_2^2 + \lambda_2\left(\tilde{\gamma}^T(G'_{\neq j\cdot})_{\cdot j}\gamma_j + G'_{jj}\gamma_j^2\right) + \lambda_1|\gamma_j| \text{ if } j \in \{1, ..., p\}, \\
\widehat{\gamma}_j &= \underset{\gamma_j}{\text{argmin}}\ (1/2)\|y - \tilde{Z}\tilde{\gamma} - Z_{\cdot j}\gamma_j\|_2^2 + \delta|\gamma_j| \text{ if } j \in \{p+1, ..., p+n\},
\end{aligned}$$

with updates

$$\begin{aligned}
\widehat{\gamma}_j &\leftarrow \frac{S\left(Z_{\cdot j}^T(y - \tilde{Z}\tilde{\gamma}) - (\lambda_2/2)\tilde{\gamma}^T(G'_{-j\cdot})_{\cdot j},\ \lambda_1/2\right)}{Z_{\cdot j}^T Z_{\cdot j} + \lambda_2 G'_{jj}} \text{ if } j \in \{1, ..., p\}, \\
\widehat{\gamma}_j &\leftarrow \frac{S\left(Z_{\cdot j}^T(y - \tilde{Z}\tilde{\gamma}),\ \lambda_1/2\right)}{Z_{\cdot j}^T Z_{\cdot j}} \text{ if } j \in \{p+1, ..., p+n\}.
\end{aligned}$$

SVM-GraphNet classification: coordinate-wise coefficient updates using infimal convolution

We can take the same approach with the SVM-GraphNet estimates, which we can write as

$$\widehat{\gamma} = \underset{\gamma}{\text{argmin}}\ (1/2\delta)\|1_{n\times 1} - Z\gamma\|_2^2 + \lambda_2\gamma_{\neq 0}^T G'\gamma_{\neq 0} + \sum_{j=0}^{p} w_j|\gamma_j| + \sum_{j=p+1}^{p+n} w_j \max(0, \gamma_j)$$

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

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

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

$$\widehat{\gamma}_j = \begin{cases} \underset{\gamma_j}{\text{argmin}}\ (1/2\delta)\|1_{N\times 1} - \tilde{Z}\tilde{\gamma} - Z_{\cdot j}\gamma_j\|_2^2 & \text{if } j = 0 \\ \underset{\gamma_j}{\text{argmin}}\ (1/2\delta)\|1_{N\times 1} - \tilde{Z}\tilde{\gamma} - Z_{\cdot j}\gamma_j\|_2^2 + \lambda_2\left(\tilde{\gamma}^T (G'_{\neq j}\cdot)_{\cdot j}\gamma_j + G'_{jj}\gamma_j^2\right) + \lambda_1|\gamma_j| & \text{if } j \in \{1, ..., p\} \\ \underset{\gamma_j}{\text{argmin}}\ (1/2\delta)\|1_{N\times 1} - \tilde{Z}\tilde{\gamma} - Z_{\cdot j}\gamma_j\|_2^2 + h(\gamma_j) & \text{if } j \in \{p+1, ..., p+n\}. \end{cases}$$

Then since

$$Z_{\cdot j} = \begin{cases} 1_{N\times 1} & \text{if } j = 0 \\ X_{\cdot j} & \text{if } j \in \{1, ..., p\} \\ e_j & \text{if } j \in \{p+1, ..., p+n\}, \end{cases}$$

we have

$$\widehat{\gamma}_j \leftarrow \begin{cases} -1_{N\times 1}^T Z_{\cdot j} + (\tilde{Z}\tilde{\gamma})^T Z_{\cdot j} + \gamma_j Z_{\cdot j}^T Z_{\cdot j} & \text{if } j = 0 \\ \frac{S\left((\tilde{Z}\tilde{\gamma})^T Z_{\cdot j} - 1_{N\times 1}^T Z_{\cdot j} - (\lambda_2/2)\tilde{\gamma}^T (G'_{\neq j}\cdot)_{\cdot j},\ \lambda_1/2\right)}{Z_{\cdot j}^T Z_{\cdot j} + \lambda_2 G'_{jj}} & \text{if } j \in \{1, ..., p\} \\ \frac{H\left((\tilde{Z}\tilde{\gamma})^T e_j - 1_{N\times 1}^T e_j, \delta\right)}{e_j^T e_j} & \text{if } j \in \{p+1, ..., p+n\}, \end{cases}$$

where yielding update:

$$\hat{\gamma}_j \leftarrow \begin{cases} \tilde{\gamma}^T \tilde{Z} 1_{N\times 1} + N(\gamma_j - 1) & \text{if } j = 0 \\ \frac{S\left((\tilde{Z}^T\tilde{\gamma} - 1_{N\times 1})^T X_{\cdot j} - (\lambda_2/2)\tilde{\beta}^T (G_{\neq j}\cdot)_{\cdot j},\ \lambda_1/2\right)}{X_{\cdot j}^T X_{\cdot j} + \lambda_2 G_{jj}} & \text{if } j \in \{1, ..., p\} \\ H\left((\tilde{Z}\tilde{\gamma})_j - 1, \delta\right) & \text{if } j \in \{p+1, ..., p+n\}. \end{cases}$$

Algorithm 1 Robust GraphNet update using infimal convolution

1. Given a set of data and parameters $\Omega = \{X, y, \lambda_1, \lambda_2\}$, previous coefficient estimates $\widehat{\alpha}^{(r)}, \widehat{\beta}^{(r)}$, and $p \times p$ positive semidefinite constraint graph $G \in S_+^{p\times p}$, let

$$\begin{aligned} \widehat{\gamma}^{(r)} &= [\widehat{\beta}^{(r)}\ \ \widehat{\alpha}^{(r)}]^T \\ Z &= [X\ \ I_{n\times n}]\,. \end{aligned}$$

2. Choose coordinate j using essentially cyclic rule 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 $\widehat{\gamma}_j^{(r)}$ using

$$\widehat{\gamma}_j^{(r+1)} \leftarrow \begin{cases} \frac{S\left(Z_{\cdot j}^T (y - \tilde{Z}\tilde{\gamma}) - (\lambda_2/2)\tilde{\gamma}^T (G'_{\neq j}\cdot)_{\cdot j},\ \lambda_1/2\right)}{Z_{\cdot j}^T Z_{\cdot j} + \lambda_2 G'_{jj}} & \text{if } j \in \{1, ..., p\} \\ S\left((y - \tilde{Z}\tilde{\gamma})_j,\ \lambda_1/2\right) & \text{if } j \in \{p+1, ..., p+n\}. \end{cases}$$

4. Repeat steps (1)-(3) cyclically for all $j \in \{1, \ldots, p+n\}$ until convergence.

Algorithm 2 SVM GraphNet classification update using infimal convolution

```
1. Given a set of data and parameters $\Omega = \{X, y, \lambda_1, \lambda_2\}$, previous coefficient estimates $\widehat{\alpha}^{(r)}, \widehat{\beta}_0^{(r)}, \widehat{\beta}^{(r)}$,
   and $p \times p$ positive semidefinite constraint graph $G \in S_+^{p \times p}$, let
       $$\begin{aligned} \widehat{\gamma}^{(r)} &= [\widehat{\beta}_0^{(r)} \;\; \widehat{\beta}^{(r)} \;\; \widehat{\alpha}^{(r)}]^T \\ Z &= \left[ y^T [1_{n\times 1} \;\; X] \;\; I_{n \times n} \right]. \end{aligned}$$
2. Choose coordinate $j$ using essentially cyclic rule 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 $\widehat{\gamma}_j^{(r)}$ using
       $$\widehat{\gamma}_j^{(r+1)} \leftarrow \begin{cases} \tilde{\gamma}^T \tilde{Z} 1_{n\times 1} + N(\gamma_j - 1) & \text{if } j = 0 \\ \frac{S\left( (\tilde{Z}^T \tilde{\gamma} - 1_{n\times 1})^T X_{\cdot j} - (\lambda_2/2) \tilde{\beta}^T (G_{\neq j \cdot})_{\cdot j}, \; \lambda_1/2 \right)}{X_{\cdot j}^T X_{\cdot j} + \lambda_2 G_{jj}} & \text{if } j \in \{1, ..., p\} \\ H\left( (\tilde{Z}\tilde{\gamma})_j - 1, \; \delta \right) & \text{if } j \in \{p+1, ..., p+n\}. \end{cases}$$
4. Repeat (1) -(3) cyclically for all $j \in \{1, \ldots, p+n\}$.
```

Parameter grid used in cross-validation

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

$$\begin{aligned} \lambda_1 &\in \{10, 11, \ldots, 99\} \\ G &\in \left\{ L, \; L + \eta I, \; L + 10^2 \eta I, \; L + 10^3 \eta I, \; L + 10^4 \eta I \right\} \text{ where } \eta = 1/\lambda_G \text{ for } \lambda_G > 0 \text{ and 1 otherwise} \\ \lambda_G &\in \{0, 10^1, 10^2, 10^3, 10^4, 10^5\} \\ \delta &\in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 1, 2, 10, 100\} \\ \lambda_1^* &\in \{1, 10^{-1}, 10^{-2}\}. \end{aligned}$$

The linear SVM was fit over parameters

$$C \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{-0}, 2, 3, 4, 5, 6, 7, 10^1, 10^2, 10^3\}. \tag{32}$$

References

References

Adler, R. J., Taylor, J. E., 2000. Random Fields and Geometry.

Allen, G., Grosenick, L., Taylor, J. E., Submitted. A Generalized Least Squares Matrix Decomposition. arXiv preprint.

Belkin, M., Niyogi, P., 2008. Towards a theoretical foundation for laplacian-based manifold methods. J Computer and Systems Sci 74, 1289–1308.

Belkin, M., Niyogi, P., Sindhwani, V., 2006. On manifold regularization. J Machine Learning Research 7, 2399–2434.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization.

Bray, S., Chang, C., Hoeft, F., 2009. Applications of multivariate pattern classification analyses in developmental neuroimaging of healthy and clinical populations. Frontiers in Human Neuroscience 3 (32), 1–12.

Breiman, L., Ihaka, R., 1984. Univ of California at Berkeley Technical report: Nonlinear discriminant analysis via scaling and ACE.

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.

Candes, E. J., Wakin, M. B., Boyd, S. P., 2003. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. PNAS 100 (5), 2197–2202.

Candes, E. J., Wakin, M. B., Boyd, S. P., 2008. Enhancing sparsity by reweighted l1 minimization. J Fourier Anal Appl 14, 69–82.

Carroll, M., Cecchi, G., Rish, I., Garg, R., Jan 2009. Prediction and interpretation of distributed neural activity with sparse models. Neuroimage.

Chappell, M., Groves, A., Whitcher, B., Woolrich, M. W., 2009. Variational bayesian inference for a nonlinear forward model. IEEE Trans Signal Processing 57 (1), 223–236.

Clemmensen, L., Hastie, T., Witten, D., Ersboll, B., In Press. Sparse Discriminant Analysis. Technometrics.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20(3), 273–297.

Cox, R. W., 1996. Afni: Software for analysis and visualization of functional magnetic resonance images. Computers in Biomedical Research 29, 162–173.

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.

De Mol, C., De Vito, E., Rosasco, L., Jan 2009. Elastic-net regularization in learning theory. Journal of Complexity.

der Kooij, A. V., 2007. Prediction accuracy and stability of regresssion with optimal scaling transformations. Technical report, Dept. Data Theory, Leiden Univ.

Donoho, D., 2006. For most large underdetermined systems of linear equations, the minimal ell-1 norm solution is also the sparsest solution. Communications on Pure and Applied Mathematics 59 (6), 797–782.

Etzel, J. A., Gazzola, V., Keysers, C., 2009. An introduction to anatomical roi-based fmri classification analysis. Brain Research 1282, 114–125.

Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association 96 (456), 1348–1360.

Friedman, J., H. T., Tibshirani, R., 2010. Regularization paths for generalized linear models via coordinate descent. J Stat Software.

Friedman, J., Hastie, T., Höfling, H., Tibshirani, R., 2007a. Pathwise coordinate optimization. Annals of Applied Statistics 1 (2), 302–332.

Friedman, J. H., 1997. On bias, variance, 0/1–loss, and the curse-of-dimensionality. Data Mining and Knowledge Discovery 1, 55–77.

Friedman, J. H., Hastie, T., Hofling, H., Tibshirani, R., 2007b. Pathwise coordinate optimization. Annals of Applied Statistics 1 (2), 302–332.

Friston, K., Holmes, A., Worsley, K., Poline, J., Jan 1995. Statistical parametric maps in functional imaging: a general linear approach. Hum Brain Mapp.

Glover, G. H., Law, C. S., 2001. Spiral in/out bold fmri for increased snr and reduced susceptibility artifacts. Magnetic Resonance in Medicine 46, 512–522.

Grosenick, L., Anderson, T., Smith, S. J., In Press. Elastic Source Selection for in vivo imaging of neuronal ensembles. Biomedical Imaging: From Nano to Macro, 6th IEEE International Symposium on.

Grosenick, L., Greer, S. M., Knutson, B., Dec 2008. Interpretable classifiers for FMRI improve prediction of purchases. IEEE Transactions on Neural Systems and Rehabilitation Engineering 16 (6), 539–548.

Grosenick, L., Klingenberg, B., Greer, S., Taylor, J., Knutson, B., 2009. Whole-brain sparse penalized discriminant analysis for predicting choice. Neuroimage (Supplement 1), S58.

Grosenick, L., Klingenberg, B., Knutson, B., Taylor, J., 2010. Interpretable multivariate models for whole-brain fMRI. Neuroimage (OHBM 2010).

Guyon, I., W. J. B. S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422.

Hanke, M., Halchenko, Y., Sederberg, P., Jan 2009. PyMVPA: A Python toolbox for multivariate pattern analysis of fMRI data. Neuroinformatics.

Hare, T. A., 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), 5623–30.

Hastie, T., Buja, A., Tibshirani, R., Jan 1995. Penalized discriminant analysis. The Annals of Statistics.

Hastie, T., Tibshirani, R., Buja, A., 1994. Flexible Discriminant Analysis by Optimal Scoring. Journal of the American Statistical Association 89 (428), 1255–1270.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data mining, Inference, and Prediction.

Haynes, J., Rees, G., Jan 2006. Decoding mental states from brain activity in humans. Nature Reviews Neuroscience.

Haynes, J., Sakai, K., Rees, G., Gilbert, S., Jan 2007. Reading hidden intentions in the human brain. Current Biology.

Hoerl, A. E., Kennard, R. W., 1970. Ridge regression: Applications to nonorthogonal problems. Technometrics 12 (1), 69–82.

Huber, P. J., E. M. R., 2009. Robust statistics.

Hutchinson, R. A., Niculescu, R., 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.

Jenatton, R., Audibert, J.-Y., Bach, F., 2011. Structured variable selection with sparsity-inducing norms. arXiv preprint.

Jia, J., Yu, B., 2009. On model selection consistency of the Elastic Net when p » n. Technical Report, University of California, Berkeley, Department of Statistics.

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.

Knutson, B., Greer, S. M., 2008. Anticipatory affect: neural correlates and consequences for choice. Philos Trans R Soc Lond B Biol Sci 363 (1511), 3771–86.

Knutson, B., Rick, S., Wimmer, G. E., Prelec, D., Loewenstein, G., 2007. Neural predictors of purchases. Neuron 53, 147–156.

Lehmann, E. L., Casella, G., 1998. Theory of Point Estimation.

Leng, C., 2008. Sparse optimal scoring for multiclass cancer diagnosis and biomarker detection using microarray data. Computational biology and chemistry 32, 417–425.

Li, C., Li, H., 2008. Network-constrained regularization and variable selection for analysis of genomic data. Bioinformatics 24 (9), 1175–1182.

Li, Y., Namburi, P. ., Yu, Z., Guan, C., Feng, J., Gu, Z., 2009. Voxel Selection in fMRI Data Analysis Based on Sparse Representation. IEEE transactions on bio-medical engineering.

McCoy, A. N., C. J. C., Haghighian, 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), 1031–40.

Mitchell, T. M., Hutchinson, R., Niculescu, R., Jan 2004. Learning to decode cognitive states from brain images. Machine Learning.

Mourão-Miranda, J., Friston, K., Brammer, M., Jan 2007. Dynamic discrimination analysis: A spatial–temporal SVM. Neuroimage.

Mourao-Miranda, J., Friston, K. J., Brammer, M., 2007. Dynamic discrimination analysis: A spatial-temporal svm. NeuroImage 36, 88–99.

Norman, K., Polyn, S., Detre, G., Haxby, J., Jan 2006. Beyond mind-reading: multi-voxel pattern analysis of fMRI data. Trends in Cognitive Sciences.

O'Toole, A., Jiang, F., Abdi, H., Pénard, N., Jan 2007. Theoretical, statistical, and practical perspectives on pattern-based classification. Journal of Cognitive Neuroscience.

Peelen, M., Fei-Fei, L., Kastner, S., Jan 2009. Neural mechanisms of rapid natural scene categorization in human visual cortex. Nature.

Pereira, F., Mitchell, T. M., Botvinick, M., Jan 2009. Machine learning classifiers and fMRI: a tutorial overview. Neuroimage.

Polyn, S., Natu, V., Cohen, J., Norman, K., Jan 2005. Category-specific cortical activity precedes retrieval during memory search. Science.

Ravikumar, P., Vu, V. Q., Yu, B., Naselaris, T., Kay., K. N., Gallant, J. L., 2009. Nonparametric sparse hierarchical models describe V1 fMRI responses to natural images. Advances in Neural Information Processing Systems 21.

Rockafellar, T. J., 1970. Convex analysis.

Rudin, L. I., Osher, S., Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. Physica D 50, 259–268.

Ryali, S., Supekar, K., Abrams, D. A., Menon, V., 2010. Sparse logistic regression for whole-brain classiÞcation of fMRI data. NeuroImage 51 (2), 752–764.

Shinkareva, S. V., Mason, R. A., Malave, V. L., Wang, W., Mitchell, T. M., Just, M. A., 2008. Using fMRI brain activation to identify cognitive states associated with perception of tools and dwellings. PLoS ONE 3 (1).

Slawski, M., zu Castell, W., Tutz, G., 2010. Feature selection guided by structural information. Ann App Stats 4 (2), 1055–1080.

Stein, C., 1956. Inadmissibility of the usual estimator for the mean of a multivariate normal distribution. Proc. Third Berkeley Symp. on Math. Statist. and Prob., Vol. 1, 197–206.

Tibshirani, R., 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B 58 (1), 267–288.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K., 2005. Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society. Series B 67 (1), 91–108.

Tikhonov, A., 1943. On the stability of inverse problems.

Tseng, P., 2001. Convergence of block coordinate descent method for nondifferentiable maximation. J. Opt. Theory Appl. 109, 474–494.

van Gerven, M. A. J., Cseke, B., de Lange, F. P., Heskes, T., 2010. Efficient Bayesian multivariate fMRI analysis using a sparsifying spatio-temporal prior. Neuroimage, 150–161.

Vogelstein, J. T., J., V., Priebe, C. E., In Press. Are mental properties supervenient on brain properties? Nature Scientific Reports.

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.

Wang, L., Zhu, J., Zou, H., 2008. Hybrid Huberized Support Vector Machines for Microarray Classification and Gene Selection. Bioinformatics 24 (3), 412–419.

Wipf, D., Nagarajan, S., 2008. A new view of automatic relevance determination. Advances in Neural Information Processing Systems 20.

Witten, D. M., Tibshirani, R., In Press. Penalized classification using Fisher's linear discriminant. Journal of the Royal Statistical Society, Series B.

Worsley, K. J., Taylor, J. E., Tomaiuolo, F., Lerchb, J., 2004. Unified univariate and multivariate random field theory. Neuroimage 23, S189–S195.

Yamashita, O., Sato, M., Yoshiika, T., Tong, F., Kamitani, Y., 2008. Sparse estimation automatically selects voxels relevant for the decoding of fmri activity patterns. Neuroimage 42 (4), 1414–1429.

Zhou, S., van de Geer, S., Bühlmann, P., Jan 2011. Adaptive lasso for high dimensional regression and gaussian graphical modeling. Electronic Journal of Statistics 5, 688–749.

Zou, H., 2006. The adaptive lasso and its oracle properties. J Am Stat Assoc 101 (467), 1418–1429.

Zou, H., Hastie, T., 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society. Series B 67 (2), 301–320.

Zou, H., Zhang, H., Jan 2009. On the adaptive elastic-net with a diverging number of parameters. The Annals of Statistics.

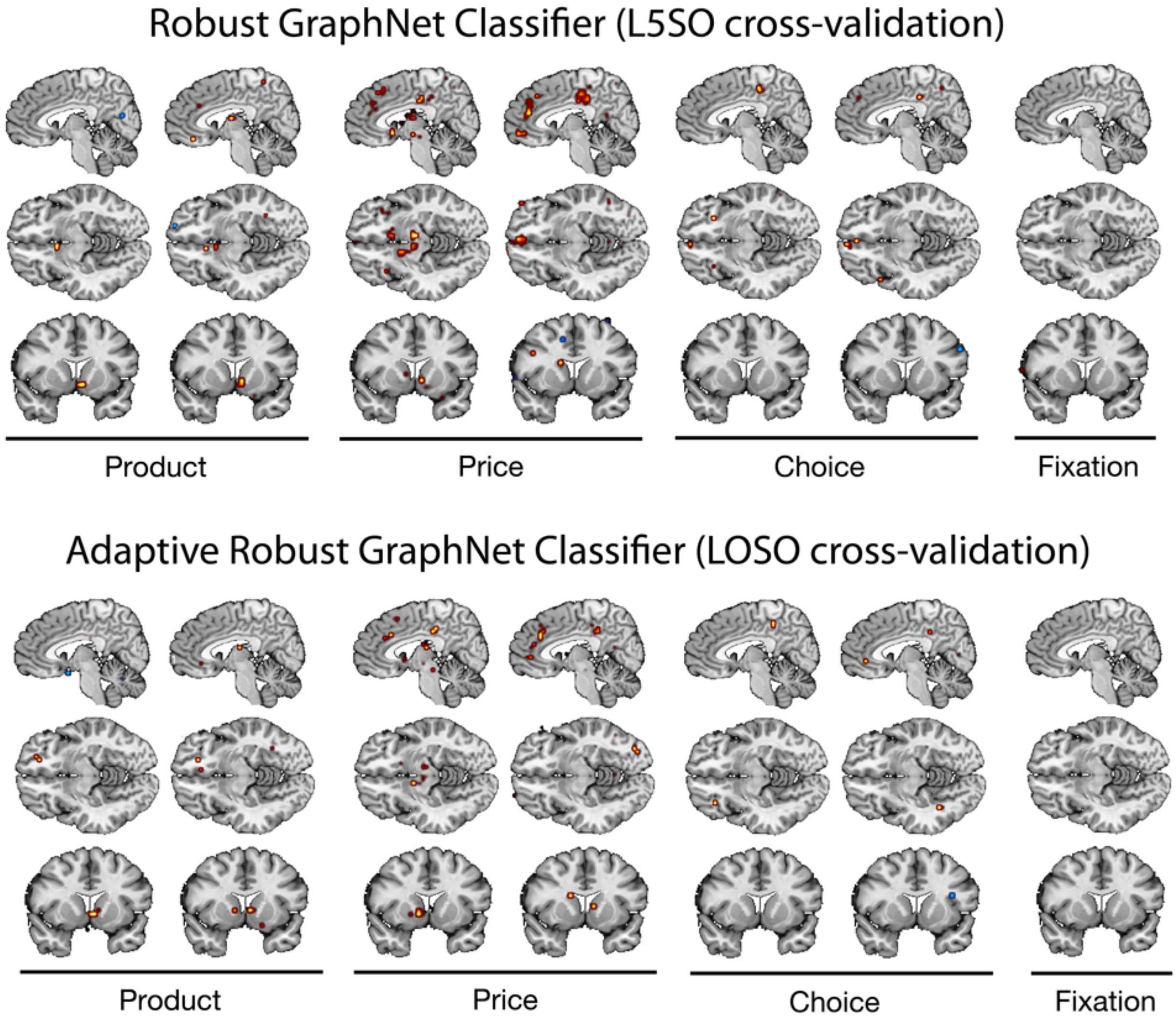
Robust GraphNet Classifier (L5SO cross-validation)
Product
Price
Choice
Fixation
Adaptive Robust GraphNet Classifier (LOSO cross-validation)
Product
Price
Choice
Fixation

Figure 5. Whole-brain classification results from the SHOP task (see Figure 3 for task structure). (Top) Median coefficient maps from the best robust GraphNet classifier fit using Leave-5-subject-out (L5SO) cross-validation are shown at two time points during each for product period, price period, and choice, period, 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 include the bilateral nucleus accumbens (NAcc) and the mesial prefrontal cortex (MPFC), as suggested by previous studies (Knutson et al. 2007; Grosenick et al. 2008), but also show the VTA and posterior cingulate to positively predict purchasing. (Bottom) Similar plots for the adaptive robust GraphNet classifier fit using leave-one-subject (LOSO) cross-validation.

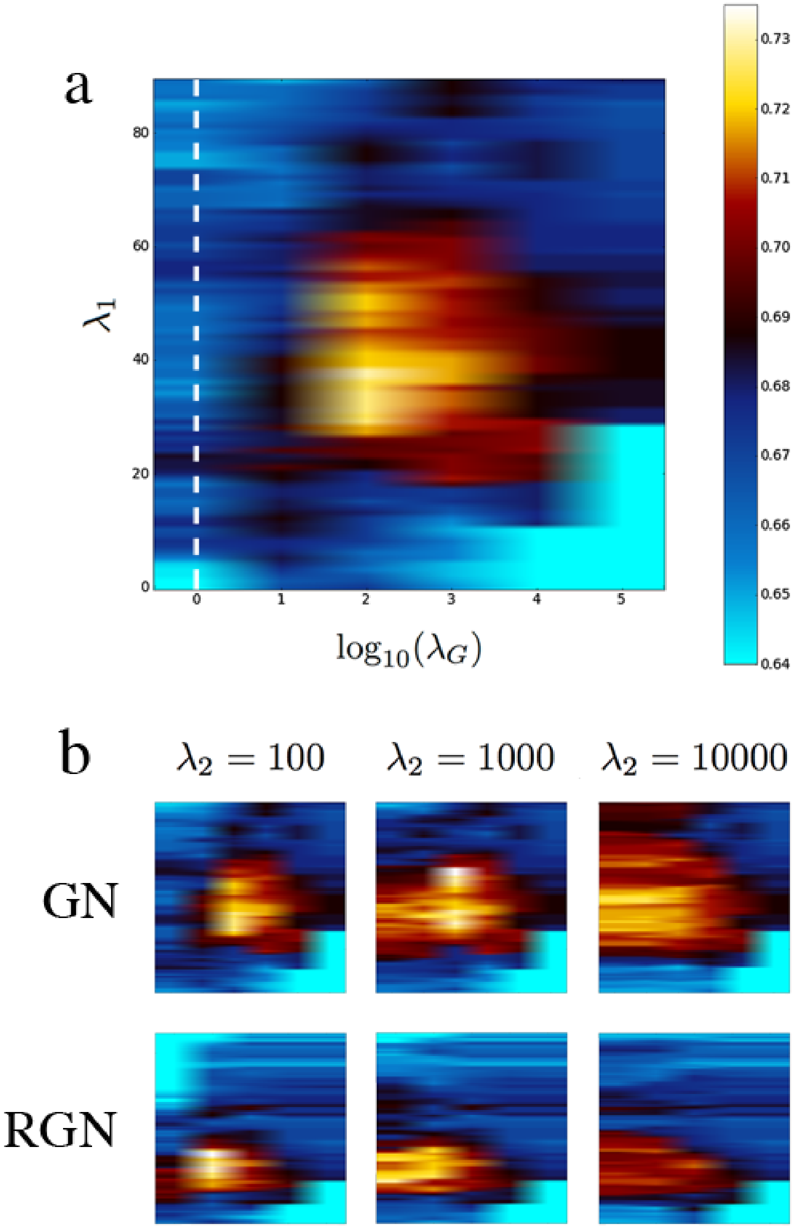

Figure 6. Median test accuracy as a function of model parameters. (a) shows a smoothed image of the median test accuracy of the GraphNet SPDA classifier (GN) as functions of hyperparameters λ_1 and λ_G (with $\lambda_2 = 0$). Warm colors indicating median classification accuracy above 70% (for L5SO cross-validation) and cool colors median accuracy below 70% (see colorbar for scale). The dotted line indicates the standard lasso solution at $\lambda_G = 0$. There is a clear maxima at $\lambda_1 = 40,\ \lambda_G = 100$. (b) shows similar plots for the GraphNet SPDA classifier (GN) and Robust GraphNet SPDA classifier (RGN) at three values of the graph diagonal scale λ_2.