Regularized Shared Response Models for fMRI

Jeremy Cohen
Department of Computer Science
Princeton University

Advisor: Professor Ken Norman
Princeton Neuroscience Institute

This thesis represents my own work in accordance with university regulations.

Contents

1 Acknowledgements

Thanks to Kiran Vodrahalli and Conor McGrory for speaking with me about this thesis with me over the course of the year.

Thanks to Professors Amirali Ahmadi, Sanjeev Arora, Elad Hazan, Han Liu, and especially Peter Ramadge for answering my questions.

Thanks to Professor Ken Norman for allowing me to engage in "unsupervised learning."

Thanks also to Adam and Zach for rooming with me, to Eve for scheming with me, to Marlis for tolerating me, to Conor for making me crack up, and to Lucas for nothing.

2 Introduction: Shared Response Models

Some time around 2004, ten Princeton students trekked over to Green Hall and laid down inside an fMRI machine. Inside the scanner, the students did typical fMRI experiment stuff, like looking at a bunch of pictures of faces, dogs, and shoes, and the like. But they also did something unusual: they watched the entirety of the movie "Indiana Jones and the Raiders of the Lost Ark" while having their brain scanned.

Several years earlier, a similar fMRI experiment [15] had established that the ventral temporal (VT) cortex of the brain plays a leading role in visual object recognition. In that experiment, scientists had used the activity of voxels in the VT cortex to "decode" which image category a subject was currently looking at. If the activity of these voxels carries information discriminative of object category, so the reasoning goes, then the region must be involved in visual object recognition. Neuroscientists increasingly use these kinds of "neural decoding" experiments to test hypotheses regarding the location of various cognitive processes within the brain [31].

One issue with this approach is that fMRI experiments typically yield very few data points, relative to the dimensionality of the data ($n << p$). For example, in [16], researchers collected from each subject just eight examples from each of the seven image categories. Each subject has at least 2,900 voxels in his/her VT cortex. Training a classifier on a seven-way multiclass prediction problem in 2900 dimensions based on just 56 examples is a tall order.

If, instead of training a separate classifier for each subject, the data across all ten subjects were pooled to train one joint classifier, then it would have 560 examples to train on. With more data, the joint classifier could pick up on fainter patterns than before, potentially enabling new neuroscientific discoveries.

Unfortunately, pooling together fMRI data across multiple subjects is anything but simple, since each human brain has a different "functional topography": a given voxel does not serve the same function in your brain that it does in mine. For example, if voxel #16 in my brain lights up whenever I look at a picture of a chair, in your brain that role might belong to voxel #18 instead. It's as if my brain speaks in German and your brain speaks in French. So long as our brains speak different languages, a single machine learning classifier could never learn to read *both* of our minds.

The solution proposed in [16] and [12] is to translate everybody's thoughts into a kind of neural Esperanto. If we somehow obtained a German-Esperanto dictionary and a French-Esperanto dictionary, then we could translate both our our brain scans into this common language, and then train a classifier to read Esperanto thoughts.

In other words, if we could map each subject's fMRI activity into a common, low-dimensional representational space, then data could be sensibly pooled across subjects. In this common space, the state of a subject's brain at any instant in time is described by the activity of a few *components* rather than by the activity of many voxels. The classifier could learn to recognize patterns in the activity of these components, which mean the same thing across subjects, rather than patterns in the activity of voxels, which don't.

One question remains: how could we learn the mappings from each subject's idiosyncratic voxel space into the common representational space (the "German-Esperanto dictionary," so to speak)? This turns out to be the reason why ten Princeton students found themselves watching "Indiana Jones and the Raiders of the Lost Ark" inside an fMRI scanner. Over the course of watching an engaging full-length action movie, all of the subjects were hearing and seeing — and maybe even thinking and feeling — the same exact thing. We might imagine that the activity of their neural *components* was in synchrony over the course of the movie, even though the activity of their *voxels* was not. We could exploit this assumption to learn the shared timecourses

of the components, as well as the mapping from each subject's voxel space into the shared component space. Afterwards, we could use these mappings in a separate pattern recognition experiment over the same subjects.

We call this the *shared response problem*: given a *synchronized dataset* such as a recording of many subjects viewing the same movie, learn the *shared response* of all subjects to that stimulus in some common low-dimensional representational space, as well as a set of *mappings* from each subject's own voxel space into the common space. A mathematical model for the shared response problem is a *shared response model*.

In this thesis we focus on *linear* shared response models, in which each component in each subject is associated with a *spatial map* over the voxels. The brain activity due to a single component is the spatial map of that component weighted by its response, and subject's full brain scan is modeled as the superposition of the activity of each of the components. In the linear case, the shared response problem thus amounts to learning a single set of shared timecourses, as well as a diferent set of spatial maps for each subject.

Below, we propose and evaluate several linear SRMs. We build off of the work of [12], who defined the problem and introduced the first SRM. We found that the original SRM performs so well because it explicitly constrains the spatial maps to be orthogonal to each other, a crucial bit of regularization that keeps the components distinct and helps prevent overfitting. In this thesis we experiment with alternative kinds of regularization — specifically, we study methods for encouraging *sparsity* and *smoothness* in the spatial maps. The original orthogonality constraint does not "mix well" with these other forms of regularization because it does not satisfy a mathematical property called *convexity*. We therefore propose and study a convex version — a so-called *spectral norm* constraint. Like the orthogonality constraint, the spectral norm constraint forces the spatial maps to be distinct from one another; unlike the orthogonality constraint, when it is combined with other convex penalties the resulting optimization problem is efficiently solvable.

First, in chapter 3, we survey related work. Then, in chapter 4, we describe our modeling philosophy and give a quick overview of important concepts from optimization. We formally describe the linear SRM problem in chapter 5. In chapter 6 we describe methods for inducing sparsity, smoothness, and distinctness in the spatial maps, and in chapter 7 we visualize the effect of each of these regularizers. We run experiments on fMRI data in chapter 8 and we conclude in chapter 9. Proofs and long derivations are deferred to the appendix.

3 Related Work

[17] showed that the brain patterns of different subjects watching the same movie exhibit a striking degree of synchrony, both in visual and auditory areas as well as in higher-level "association cortices," which are the regions of the brain thought to house the processing that converts sensory "inputs" into motor "outputs."

Exploiting this finding, [34] proposed an algorithm for the functional alignment of fMRI data from multiple subjects who have been presented with the same movie stimulus, under the assumption that the subjects experienced a synchronized response. The idea is to learn a "rubber-sheet warping" from each subject's individual brain topography into the common space. [16] proposed a different method, called *hyperalignment*, for the same problem. Hyperalignment learns a square orthogonal rotation matrix to transform each subject's individual brain topography into the common space. In hyperalignment, the dimension of the common space is the same as the dimension of the individual space (i.e. the number of voxels). However, each "hypervoxel" does not correspond to any particular location in physical space — rather, it corresponds to the weighted activity of a number of voxels which may not be anywhere near each other. Since the dimensionality of the common space is so large, after running hyperalignment [16] did a dimensionality reduction using PCA before further analysis. The method proposed in [12], called the *shared response model* (SRM), obviates the need for this hacky step by building a dimensionality reduction straight into the model: SRM learns a

matrix with a small number of orthogonal rows to transform each subject's individual space into the common space. In SRM, the dimensionality of the common space is small — the same as the number of rows in the transformation matrix. Each of these rows can be interpreted as a spatial map over the voxels. [12] also proposed a probabilistic variant of SRM which models noise and covariance, and which was shown to outperform the basic SRM.

Moving beyond the assumption of a shared response into the more general case in which the timecourses of different subjects are not necessarily linked, we should remark that a whole slew of matrix factorization methods have been developed to decompose fRMI data into a small number of latent components, each consisting of a timecourse of activations and a spatial map of voxel weights. The most widely used of these is spatial independent components analysis (ICA) [2, 3], which assumes that the spatial maps are statistically independent, i.e. that for each voxel v the weight of map 1 at voxel v is independent of the weight of map 2 at voxel v, and so on. [13] speculated that the reason why ICA works so well is not because it assumes independence, but rather because as a side effect it tends to produce sparse spatial maps. [9] disputed this claim, but nevertheless a number of methods have been proposed which assume sparsity in the spatial maps rather than statistical independence. For example, [38, 37] use an ℓ_1 penalty to encourage sparse spatial maps, with an additional spatial-smoothness-inducing penalty that apparently improves performance when voxels are correlated.

Some methods make stronger assumptions about the nature of the spatial maps. Topographic Factor Analysis (TFA) [30] assumes that each spatial map is a spherical radial basis function, while PrAGMATiC [18] assumes that each spatial map is a convex tile on the cortical surface.

A Princeton senior thesis [40] proposed a matrix factorization method which assumes sparsity in the *time-courses* rather than the spatial maps, implying that at each point in time only a few neural components should be active.

4 Convex Optimization

In this section we describe and justify our "optimization-first" modeling framework, and then give a primer on convex optimization.

4.1 Optimization Models

Suppose that we would like to "learn" a latent object (say, a set of timecourses and spatial maps) based on observed data. In a Bayesian setting, one might encourage the latent object to possess a certain property by assuming that the object was "drawn" from a "prior distribution" which favors that property. For example, one might use zero-mean Gaussian prior to favor small vectors, or a "Laplace" prior to favor sparse vectors. A Bayesian would then try to compute via sampling (or approximate via variational inference) the "posterior distribution" of the latent object — that is, the conditional probability, given the data, that the latent object takes on every possible value. In our case, the posterior distribution would be a function that takes in "possible spatial maps" and spits out "probability, given the subject's fMRI data, that these are the subject's spatial maps." Of course, there are an infinite number of "possible spatial maps," and so it is perhaps not surprising that computing this posterior distribution is a massive headache. The good news is that we don't *care* about, say, the probability that subject four's sixth spatial map is shaped like Elvis Presley. All we need to know is: what are subject four's *most likely* spatial maps, given the data. Bayesians call this "MAP inference," and it just boils down to optimizing a so-called "joint likelihood" function which measures both the probability that

the data was generated given the latent object, and the probability that the latent object was generated from the prior. Optimizing the former term ensures that the latent object fits the data well; optimizing the latter term ensures that the latent object possess the desired properties that were encoded into its prior.

Since we do not need a full posterior distribution over spatial maps, and since we have no reason to believe that the spatial maps were "generated" from any particular distribution, we will formulate our SRM models directly as optimization problems rather than as probabilistic generative models. In this optimization-first framework [8], we encourage the spatial maps to possess various properties such as distinctness, sparsity, or smoothness by imposing *constraints* or by adding *penalties* to the optimization problem. We especially like constraints and penalties that satisfy a mathematical property called *convexity*, since efficient algorithms have been developed to solve optimization problems with convex constraints and penalties.

In this section we review the basics of convex optimization, with an eye towards giving sufficient background for the models and algorithms presented later in this thesis. We define convex sets, convex functions, and convex optimization problems, and then we discuss convex relaxation, a general strategy for turning difficult optimization problems into easy ones.

4.2 Convex Sets

A set is convex if, for every pair of points in the set, the line segment between those two points lies entirely within the set.

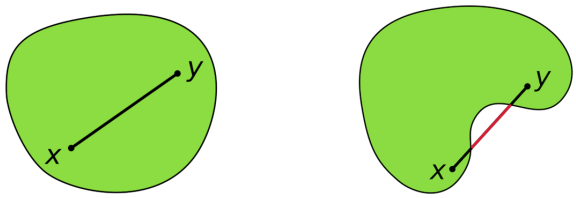


Figure 1: A convex set (left) and a non-convex set (right) (Wikipedia)

For example, the set on the left is convex, but the set on the right is not, because it contains two points that are connected by a line segment that lies partially outside of the set.

Figure 2: More convex sets (Amirali Ahmadi)

4.3 Convex Functions

A function f is convex if, for any two points x and y, the straight line connecting $f(x)$ and $f(y)$ lies entirely above f at all values in the interval $[x, y]$. Convex functions are "bowl-shaped."

Figure 3: More non-convex sets (Amirali Ahmadi)

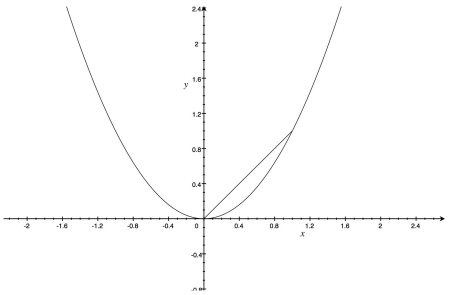
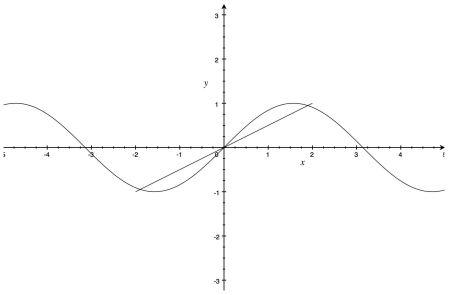


Figure 4: A convex function (top) and a non-convex function (bottom). For the convex function, the line segment between any two points on the graph lies entirely above the graph of the function. For the non-convex function, the line segiment between two points on the graph at times dips below the graph.



4.4 Convex Optimization Problems

We say that an optimization problem

$$\min f(x) \quad \text{subject to} \quad x \in \mathcal{C} \tag{1}$$

is a *convex optimization problem* if f is a convex function and $\mathcal{C}$ is a convex set. This is a useful distinction to make because there exist algorithms that can efficiently solve any convex optimization problem to global optimality [6]. In contrast, when an optimization problem is non-convex, one can apply a standard optimization algorithm such as gradient descent, but there is no guarantee that the solution to which the algorithm converges is in fact a global minimum.

4.5 Convex Relaxations

Sometimes we would like to optimize a function that is convex over $\mathbb{R}^n$ over a non-convex set $\mathcal{C} \subset \mathbb{R}^n$. In such settings, generic convex optimization algorithms are not guaranteed to find a global minimum. Figure

5 illustrates this situation. Suppose that we would like to find the point in the green-colored region of the number line (a non-convex set) that minimizes the function plotted above. If we run, say, a gradient descent algorithm starting from an initial point in the middle green interval, then the algorithm will converge to the correct global minimum. However, if we instead are unlucky enough that we initialize our search at a point in the left or the right green interval, then gradient descent will only converge to a local minimum, since gradient descent is a fearful algorithm that will never venture beyond the interval that it starts in.

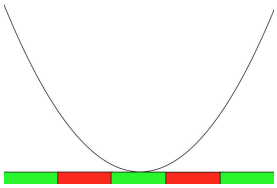


Figure 5: Finding the minimum of this function over the green regions of the number line is a challenging optimization problem, since the feasible set is not convex. For example, the gradient descent optimization algorithm, when initialized to a point in one of the "flanking" green regions, will never reach the global minimum.

How, then, can we ever hope to optimize a function over a nonconvex set?

1. In some special cases, the optimization problem will happen to have a closed form solution. This turns out to be the case for the original orthogonal SRM.
2. We could run a convex optimization algorithm anyway and pray to the optimization deity of our choice, such as Yahweh, Vishnu, or Yuri Nesterov, that it converges to a global minimum and not a local one.
3. We could "relax" the problem and optimize over a convex set instead. For example, if $\mathcal{C}$ is our non-convex set, we could optimize over the *convex hull* of $\mathcal{C}$, which is the smallest convex set that completely contains $\mathcal{C}$. Figure 6 depicts a non-convex set (a five-point star) and its convex hull (a pentagon). Intuitively, the convex hull of $\mathcal{C}$ is what you get when you place a rubber band around $\mathcal{C}$.

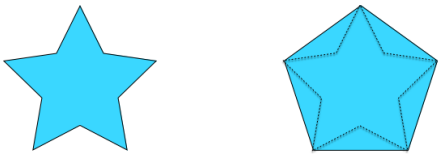


Figure 6: A non-convex set (left) and its convex hull (right). Intuitively, the convex hull of a set $\mathcal{C}$ is the set enclosed by a rubber band that is stretched around $\mathcal{C}$.

> In this thesis, I "relax" an optimization problem over the non-convex set of orthogonal matrices into an optimization problem over the convex set of matrices with bounded spectral norm. Even though the solution to the relaxed problem will not necessarily be exactly orthogonal, the idea is that its rows will be *sort of* orthogonal, which is hopefully good enough.

5 Linear SRMs

In this section we formulate the linear SRM problem as an optimization problem in two variables — the shared timecourses and the subject-specific spatial maps — and we show how to optimize over the timecourses.

Let $\{\mathbf{Y}_i\}_{i=1}^S$ be a synchronized dataset collected from S subjects, where each $\mathbf{Y}_i \in \mathbb{R}^{N \times V_i}$ is the fMRI data from the i-th subject, N is the number of TRs in the dataset (e.g. the length of the movie, in fMRI frames), and V_i is the number of voxels in the i-th subject.

A linear SRM with K components assumes that these data were generated based on a shared response $\mathbf{W} \in \mathbb{R}^{N \times K}$ and a set of subject-specific spatial maps $\{\mathbf{M}_i\}_{i=1}^S$, with each $\mathbf{M}_i \in \mathbb{R}^{K \times V_i}$.

Specifically, a linear SRM assumes that:

$$\mathbf{Y}_i \approx \mathbf{W}\mathbf{M}_i$$

To "solve" an *unregularized* SRM is to find a shared response $\mathbf{W}$ and a set of subject-specific spatial maps $\mathbf{M}_1 \dots \mathbf{M}_S$ that minimize some loss function which measures how far away the "reconstruction" $\mathbf{W}\mathbf{M}_i$ is from the i-th subject's actual data $\mathbf{Y}_i$. In this thesis we always work with the squared ℓ_2 loss, in which case solving an unregularized SRM boils down to solving the following optimization problem:

$$\underset{\mathbf{W},\{\mathbf{M}_i\}_{i=1}^S}{\arg\min} \quad \frac{1}{2}\sum_{i=1}^{S} ||\mathbf{Y}_i - \mathbf{W}\mathbf{M}_i||_F^2 \tag{2}$$

Incidentally, solving an unregularized SRM is equivalent to finding the maximum a posteriori estimate for $\mathbf{W}$ and $\{\mathbf{M}_i\}_{i=1}^S$ under a Gaussian probability model:

$$\mathbf{Y}_i \sim \mathcal{N}\left(\mathbf{W}\mathbf{M}_i, \sigma^2\mathbf{I}\right) \tag{3}$$

Most of the SRMs studied in this thesis introduce regularization on the spatial maps in the form of a penalty function $\Omega(\mathbf{M}_i)$. This will encourage us to learn spatial maps $\mathbf{M}_i$ for which $\Omega(\mathbf{M}_i)$ is *low*.

Additionally, we will constrain the Euclidean norm of each column of $\mathbf{W}$ to be less than one, as otherwise the problem is unidentifiable with respect to relative scalings of the timecourses and spatial maps.

The regularized SRM problem is:

$$\underset{\mathbf{W},\{\mathbf{M}_i\}_{i=1}^S}{\arg\min} \quad \frac{1}{2}\sum_{i=1}^{S} ||\mathbf{Y}_i - \mathbf{W}\mathbf{M}_i||_F^2 + \Omega(\mathbf{M}_i) \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1 \tag{4}$$

Note that this framework allows for hard constraints to be imposed on $\mathbf{M}_i$ by taking $\Omega(\mathbf{M}_i)$ to be the indicator function $\mathbf{1}_{\mathcal{C}}$ of a set $\mathcal{C}$, i.e. 0 when $\mathbf{M}_i$ is in $\mathcal{C}$ and $+\infty$ otherwise.

Problem (4) is nonconvex, and therefore difficult to solve to global optimality. However, with either $\mathbf{W}$ or $\{\mathbf{M}_i\}_{i=1}^S$ held fixed, it is convex in the other variable. We will therefore approach (4) with a heuristic called *alternating minimization* (a.k.a. "block coordinate descent"). Alternating minimization alternates between solving for $\mathbf{W}$ and solving for $\{\mathbf{M}_i\}_{i=1}^S$.

When $\mathbf{W}$ is held fixed, the objective function of (4) decomposes into a sum of terms, one for each variable $\mathbf{M}_i$; therefore, step 3 may be carried out in parallel over the subjects.

Algorithm 1 Alternating Minimization for SRM

Initialize each $\mathbf{M}_i$ randomly.
for iter = 1 to num iters **do**
  Hold all the $\mathbf{M}_i$ fixed and solve for $\mathbf{W}$
  Hold $\mathbf{W}$ fixed and solve for each of the $\mathbf{M}_i$.

Therefore, to carry out alternating minimization for regularized SRM we need to know how to solve the following two optimization problems:

$$\arg\min_{\mathbf{W}} \quad \sum_{i=1}^{S} ||\mathbf{Y}_i - \mathbf{W}\mathbf{M}_i||_F^2 \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1 \tag{5}$$

$$\arg\min_{\mathbf{M}_i} \quad \frac{1}{2}||\mathbf{Y}_i - \mathbf{W}\mathbf{M}_i||_F^2 + \Omega(\mathbf{M}_i) \tag{6}$$

5.0.1 Learning the timecourses

We can write problem (5) in a more compact form by defining

$$\tilde{\mathbf{Y}} := \begin{bmatrix} \mathbf{Y}_1 \\ \vdots \\ \mathbf{Y}_S \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{M}} := \begin{bmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_S \end{bmatrix}$$

Then (5) is equivalent to a least-squares problem with an extra norm constraint:

$$\begin{aligned} & \arg\min_{\mathbf{W}} \quad ||\tilde{\mathbf{Y}} - \mathbf{W}\tilde{\mathbf{M}}||_F^2 \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1 \\ = & \arg\min_{\mathbf{W}} \quad ||\tilde{\mathbf{Y}}^T - \tilde{\mathbf{M}}^T\mathbf{W}^T||_F^2 \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1 \end{aligned} \tag{7}$$

We solve (7) using the Mosek commercial optimization solver. The matrices $\tilde{\mathbf{Y}} \in \mathbb{R}^{NS \times V}$ and $\tilde{\mathbf{M}} \in \mathbb{R}^{KS \times V}$ are large enough that forming them in memory is impossible. Fortunately, (7) is equivalent to

$$\begin{aligned} & \arg\min_{\mathbf{W}} \quad -2\text{Tr}(\tilde{\mathbf{M}}\tilde{\mathbf{Y}}^T\mathbf{W}) + ||\mathbf{W}\tilde{\mathbf{M}}||_F^2 \qquad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1 \\ = & \arg\min_{\mathbf{W}} \quad -2\text{Tr}(\tilde{\mathbf{M}}\tilde{\mathbf{Y}}^T\mathbf{W}) + ||\mathbf{W}\mathbf{L}||_F^2 \qquad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1 \end{aligned}$$

where $\mathbf{L}\mathbf{L}^T = \tilde{\mathbf{M}}\tilde{\mathbf{M}}^T$ is the Cholesky decomposition of $\tilde{\mathbf{M}}\tilde{\mathbf{M}}^T$. In this equivalent form, the objective function has few enough terms that any optimization package can solve it. Note that $\tilde{\mathbf{M}}\tilde{\mathbf{M}}^T$ and $\tilde{\mathbf{M}}\tilde{\mathbf{Y}}^T$ are $K \times K$ and $K \times N$ matrices which can be computed without forming $\tilde{\mathbf{M}}$ or $\tilde{\mathbf{Y}}$ in memory as:

$$\tilde{\mathbf{M}}\tilde{\mathbf{M}}^T = \sum_{i=1}^{S} \mathbf{M}_i\mathbf{M}_i^T \quad \text{and} \quad \tilde{\mathbf{M}}\tilde{\mathbf{Y}}^T = \sum_{i=1}^{S} \mathbf{M}_i\mathbf{Y}_i^T$$

Solving (7) using a generic conic programming solver like Mosek works fine for our application, since $\mathbf{W}$ is a medium-size matrix, with only $NK = 10,000$ to $50,000$ elements, and speed is not very important, since the runtime of alternating minimization is dominated by the costly learn-the-spatial-maps step. If $\mathbf{W}$ were larger or performance were more critical, we could alternatively adopt the approach proposed in [29], which is to solve the Lagrange dual of (7), itself a convex optimization problem with only K variables. This alternative approach is described in detail in Appendix A.

5.0.2 Learning the spatial maps

How to solve (6) to learn the spatial maps depends on the choice of penalty function $\Omega(\mathbf{M}_i)$. In the following section we will introduce a several penalties that promote sparsity, smoothness, and distinctness in the spatial maps. Each penalty will give rise to its own algorithm for solving (6).

Since the learn-the-spatial-maps step decomposes into a separate optimization problem for each subject, in the next section, we drop the subscript i from $\mathbf{M}_i$ and $\mathbf{Y}_i$.

6 Penalties for Brain Maps

In this section we develop several penalty functions $\Omega(\mathbf{M})$ for brain maps $\mathbf{M} \in \mathbb{R}^{K \times V}$ that induce sparsity, smoothness, and distinctness in the solution $\hat{\mathbf{M}}$ to the following optimization problem:

$$\hat{\mathbf{M}} = \arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}_i||_F^2 + \Omega(\mathbf{M}) \tag{8}$$

Recall that $\mathbf{Y} \in \mathbb{R}^{N \times V}$ is a single subject's fMRI data and that $\mathbf{W} \in \mathbb{R}^{N \times K}$ is our current guess at the shared timecourses.

In a bit of confusing notation, we will use $\mathbf{w}_k$ to denote the k-th timecourse (the k-th *column* of $\mathbf{W}$) and $\mathbf{m}_k$ to denote the k-th spatial map (the k-th *row* of $\mathbf{M}$).

6.1 No penalties

With no penalty function Ω, learning the spatial maps entails solving the ordinary least-squares problem:

$$\arg\min_{\mathbf{M}} \quad ||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 \tag{9}$$

Since this problem is convex, setting the gradient to zero and solving for $\mathbf{M}$ gives the closed-form solution

$$\hat{\mathbf{M}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{Y} \tag{10}$$

6.2 An ℓ_1 penalty for sparsity

Suppose that we would like to encourage $\mathbf{M}$ to be *sparse* — that is, to have many entries set to exactly zero. A natural way promote sparsity would be to penalize the number of nonzero elements, called the *cardinality*, of $\mathbf{M}$. Unfortunately, the cardinality is not a convex function. Therefore, we will instead penalize a popular convex surrogate for cardinality [6, p. 310], the element-wise ℓ_1 norm of $\mathbf{M}$, defined as the sum of the absolute values of the elements in $\mathbf{M}$:

$$||\mathbf{M}||_1 := \sum_{k=1}^{K}\sum_{v=1}^{V}|M_{kv}| \tag{11}$$

With an ℓ_1 penalty $\Omega(\mathbf{M}) = \alpha||\mathbf{M}||_1$, learning the spatial maps entails solving the following optimization problem:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \alpha||\mathbf{M}||_1 \tag{12}$$

where α is a parameter that controls the strength of the sparsity regularization.

6.2.1 Intuition

Why does penalizing the ℓ_1 norm of $\mathbf{M}$ encourage the solution of the optimization problem (12) to be sparse? Well, as an optimization problem in the "Lagrange" form, (12) is equivalent to an optimization problem in the "constrained" form for some choice of t:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 \quad \text{subject to} \quad ||\mathbf{M}||_1 \leq t$$

The feasible set of this constrained problem, the so-called ℓ_1 ball $\{\mathbf{x} : ||\mathbf{x}||_1 \leq t\}$, is shaped like a diamond with "corners" at points where one of the coordinates is zero:

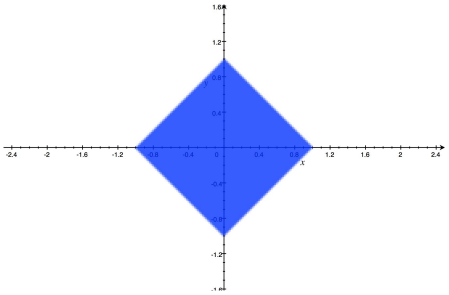


Figure 7: The ℓ_1 unit ball in two dimensions

It makes intuitive sense that the minimizer of any function over this diamond-shaped feasible set is somewhat likely to occur at at a "corner" — that is, at a sparse solution [22, p. 71].

6.2.2 Optimization

Problem (12) goes by many names — among them: ℓ_1 penalized least-squares, basis pursuit, and the lasso [36] — and has been widely studied by the statistics and machine learning communities over the past two decades. We will solve (12) using a convex optimization algorithm well-suited to non-smooth objective functions, the proximal gradient method [32].

The proximal gradient method (PGM) is an algorithm for solving convex optimization problems of the form

$$\arg\min_{\mathbf{M}} \; f(\mathbf{M}) + g(\mathbf{M})$$

where f is convex with Lipschitz-continuous gradient, and g is convex, but not necessarily differentiable.

A key computational primitive used by the proximal gradient method is the so-called *proximal operator* of g. The proximal operator of a function $g : \mathbb{R}^{m \times n} \to \mathbb{R}$ is a function $\mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ defined as [32]:

$$\mathbf{prox}_g(\mathbf{M}) := \arg\min_{\mathbf{C}} \left(g(\mathbf{C}) + \frac{1}{2}||\mathbf{M} - \mathbf{C}||_F^2 \right) \tag{13}$$

For many interesting functions g (including, as we will see, the ℓ_1 norm), the proximal operator can be computed on the cheap.

Each iteration of the proximal gradient method takes a step in the direction of the negative gradient of f, and then applies the proximal operator of g:

$$\mathbf{M}_{t+1} = \mathbf{prox}_{\eta_t g} \left(\mathbf{M}_t - \eta_t \nabla f \right)$$

where η_t is the step size in iteration t.

To solve (12), we use the splitting

$$f(\mathbf{M}) = \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 \qquad g(\mathbf{M}) = \alpha||\mathbf{M}||_1$$

The gradient of f is

$$\nabla f(\mathbf{M}) = \mathbf{W}^T\mathbf{W}\mathbf{M} - \mathbf{W}^T\mathbf{Y} \tag{14}$$

The proximal operator of $\lambda||\cdot||_1$ is often called the *soft-thresholding* operator [32, p. 188] because it sets values in $[-\lambda, \lambda]$ to zero, and shrinks all other values towards zero by λ:

$$\left(\mathbf{prox}_{\lambda||\cdot||_1}(\mathbf{M}) \right)_{ij} = \begin{cases} M_{ij} - \lambda & \text{if } M_{jj} \geq \lambda \\ 0 & \text{if } -\lambda < v_j < \lambda \\ M_{ij} + \lambda & \text{if } M_{ij} \leq -\lambda \end{cases} \tag{15}$$

6.3 An elastic net penalty for saner sparsity

ℓ_1-regularized least squares is known to have a flaw [43]: when a group of voxels is correlated, the ℓ_1 penalty will tend to zero out all but one of the voxels. This may not be the best behavior if we are looking to discover brain systems. The solution proposed in [43], known as the *elastic net*, is to add an additional squared ℓ_2 penalty. With the elastic net penalty $\Omega(\mathbf{M}) = \alpha||\mathbf{M}||_1 + \beta||\mathbf{M}||_F^2$, learning the spatial maps entails solving:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \alpha||\mathbf{M}||_1 + \beta||\mathbf{M}||_F^2 \tag{16}$$

where α is a parameter that controls the strength of the sparsity regularization, and β is a parameter that "softens" the sparsity regularization so as to permit groups of correlated voxels to pass through un-sparsified.

6.3.1 Optimization

By splitting (16) as follows, we may use the proximal gradient method:

$$f(\mathbf{M}) = \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \beta||\mathbf{M}||_F^2 \qquad g(\mathbf{M}) = \alpha||\mathbf{M}||_1 \tag{17}$$

The gradient of f is:

$$\nabla f(\mathbf{M}) = \mathbf{W}^T\mathbf{W}\mathbf{M} - \mathbf{W}^T\mathbf{Y} + 2\beta\mathbf{M} \tag{18}$$

6.4 A "smooth lasso" penalty for smoothness

A sensible neuroscientific assumption is that brain systems are likely to be concentrated in neighboring voxels. We might want to encode this assumption into our model by forcing the spatial maps to vary *smoothly* in space. In other words, a spatial map should be discouraged from taking on wildly distant values at two voxels that are neighbors on the grid. One way to express this preference as a convex penalty is to penalize the squared difference between each neighboring pair of voxels:

$$\Omega(\mathbf{m}) = \sum_{\text{voxels } i \text{ and } j \text{ are neighbors}} (m_i - m_j)^2 \tag{19}$$

There is an equivalent way to formulate (19) that is more amenable to matrix algebra. The *graph laplacian* $\mathbf{L} \in \mathbb{N}^{V \times V}$ of a brain G with V voxels is defined as the degree matrix of G minus the adjacency matrix of G:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{20}$$

The degree matrix $\mathbf{D} \in \mathbb{N}^{V \times V}$ is a diagonal matrix with the number of neighbors of each voxel on the diagonal. The adjacency matrix $\mathbf{A} \in \{0, 1\}^{V \times V}$ has a one at location (i, j) if voxels i and j are neighbors, and a zero otherwise.

For a brain with graph laplacian $\mathbf{L}$, the smoothness penalty (19) may be equivalently written as:

$$\Omega(\mathbf{m}) = \mathbf{m}^T \mathbf{L}\, \mathbf{m} \tag{21}$$

To see why this is so, notice that (19) is equivalent to:

$$\begin{aligned}
\Omega(\mathbf{m}) &= \sum_{i,j \text{ neighbors}} (m_i - m_j)^2 \\
&= \sum_{i,j \text{ neighbors}} m_i^2 - 2 m_i\, m_j + m_j^2 \\
&= \sum_i \text{deg}(i)\, m_i^2 \; - \sum_{i,j:\, \mathbf{A}_{ij}=1} m_i\, m_j \\
&= \mathbf{m}^T \mathbf{D}\, \mathbf{m} - \mathbf{m}^T \mathbf{A}\, \mathbf{m} \\
&= \mathbf{m}^T (\mathbf{D} - \mathbf{A})\, \mathbf{m} \\
&= \mathbf{m}^T \mathbf{L}\, \mathbf{m}
\end{aligned}$$

We can combine this smoothness penalty with an elastic net penalty to encourage simultaneous smoothness and sparsity. The resulting optimization problem is:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \alpha||\mathbf{M}||_1 + \beta||\mathbf{M}||_F^2 + \gamma \sum_{k=1}^{K} \mathbf{m}_k^T \mathbf{L} \mathbf{m}_k \tag{22}$$

where α and β control the sparsity, and γ controls the smoothness.

Following [37], we call this the *smooth lasso*.

6.4.1 Related work

[37] used the smooth lasso in an unsupervised dictionary learning setting much like ours, and [19] used it in a regression setting. [24] studied the statistical properties of the smooth lasso in linear regression problems.

6.4.2 Optimization

We may again apply the proximal gradient method, with the splitting:

$$f(\mathbf{M}) = \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \beta||\mathbf{M}||_F^2 + \gamma \sum_{k=1}^{K} \mathbf{m}_k^T \mathbf{L}\mathbf{m}_k \qquad g(\mathbf{M}) = \alpha||\mathbf{M}||_1$$

The gradient of f is:

$$\nabla f(\mathbf{M}) = \mathbf{W}^T\mathbf{W}\mathbf{M} - \mathbf{W}^T\mathbf{Y} + 2\beta\mathbf{M} + 2\gamma \sum_{k=1}^{K} \mathbf{M}\mathbf{L} \tag{23}$$

For a brain with 80,000 voxels, the Laplacian is an 80,000 x 80,000 matrix, so we store it as a sparse matrix and use sparse matrix operations.

6.5 An orthogonal constraint for distinctnesss

The evidence suggests that components whose spatial maps are explicitly constrained to be distinct from one another provide a more informative basis for brain activity than components whose spatial maps are not so constrained.

One easy way to ensure that the spatial maps are distinct from one another is to require that all pairs of spatial maps are orthogonal. If we additionally require that each spatial map has norm c, then we have the following constraint:

$$\mathbf{M}\mathbf{M}^T = c^2\,\mathbf{I} \tag{24}$$

(To see this, notice that the diagonal elements of $\mathbf{M}\mathbf{M}^T$ are the dot products between rows of $\mathbf{M}$, i.e. spatial maps, and themselves, while the off-diagonal elements of $\mathbf{M}\mathbf{M}^T$ are the dot products between two different spatial maps.)

Learning the spatial maps under this orthogonality constraint $\Omega(\mathbf{M}) = \mathbf{1}_{\mathbf{M}\mathbf{M}^T = c^2\mathbf{I}}$ yields the following optimization problem:

$$\underset{\mathbf{M}}{\arg\min} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 \quad \text{subject to} \quad \mathbf{M}\mathbf{M}^T = c^2\mathbf{I} \tag{25}$$

This optimization problem is not convex, as the feasible set $\{\mathbf{M} \in \mathbb{R}^{K\times V} : \mathbf{M}\mathbf{M}^T = c^2\mathbf{I}\}$ is not a convex set. Nevertheless, it turns out that problem (25) admits a simple analytic solution:

$$\hat{\mathbf{M}} = c\,\mathbf{V}\mathbf{U}^T \qquad \text{where} \quad \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{Y}^T\mathbf{W} \tag{26}$$

When $c = 1$, problem (25) is called the *orthogonal Procrustes problem*, and the proof can be found in [20, p. 601]. The extension to all $c > 0$ is trivial, and the proof is in Appendix C.

6.6 A spectral norm constraint for convex distinctness

The orthogonal constraint $\mathbf{M}\mathbf{M}^T = c\,\mathbf{I}$ certainly forces the spatial maps to be distinct from one another. However, suppose that we would like to learn spatial maps that are both distinct *and* sparse. We might think

to add an ℓ_1 penalty to problem (25); unfortunately, adding this penalty would "break" the analytical solution. Moreover, since the orthogonality constraint is not convex, the resulting optimization problem would not be solvable with convex optimization algorithms.

An alternative approach is to "relax" the orthogonality constraint by optimizing over its convex hull instead. The convex hull of the set of matrices whose rows have norm c and are pairwise orthogonal is the set of matrices with spectral norm no more than c:

$$\text{conv}(\{\mathbf{M} : \mathbf{M}\mathbf{M}^T = c^2\mathbf{I}\}) = \{\mathbf{M} : ||\mathbf{M}||_2 \leq c\} \tag{27}$$

This fact is proved in Appendix C.

The spectral norm $||\mathbf{M}||_2$ is the largest singular value of $\mathbf{M}$, or equivalently the maximum 2-norm of the product of $\mathbf{M}$ and a unit-norm vector:

$$||\mathbf{M}||_2 = \sigma_1(\mathbf{M}) = \max_{||\mathbf{v}||=1} ||\mathbf{M}\mathbf{v}||_2$$

Learning the spatial maps under this spectral norm constraint entails solving the following optimization problem:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 \quad \text{subject to} \quad ||\mathbf{M}||_2 \leq c \tag{28}$$

6.6.1 Intuition

To get some intuition for the spectral norm constraint, we ask: for which values of x and y does the matrix

$$\mathbf{M} = \begin{bmatrix} 1 & 0 \\ x & y \end{bmatrix}$$

satisfy the spectral norm bound $||\mathbf{M}||_2 \leq c$ for values of c near 1? Under the corresponding orthogonality constraint $\mathbf{M}\mathbf{M}^T = \mathbf{I}$, the only allowable values would be $[x, y] = [0, 1]$ or $[0, -1]$, which are maximally distinct from the filled-in row of $\mathbf{M}$. Clearly, the spectral norm constraint will permit a wider range of values for $[x, y]$. The question is: just how distinct from $[1, 0]$ are the vectors $[x, y]$ that satisfy the spectral norm constraint?

Taking advantage of the closed-form formula for the singular values of a 2x2 matrix, in figure 8 we plot the 1.1 level set, the 1.01 level set, and the 1.001 level set of the spectral norm of $\mathbf{M}$. The region inside the c-level set is the feasible set of the constraint $||\mathbf{M}||_2 \leq c$. One can see that when $c = 1.1$, very little distinctness regularization is occurring, since even the vector $[x, y] = [1, 0]$ would be permitted. When the constraint is tightened to $c = 1.01$, the "worst" vector that would be permitted (that is, the least distinct from the filled-in row of $\mathbf{M}$) is $[0.3, 0]$. When the constraint is further tightened to $c = 1.001$, virtually the only vectors $[x, y]$ that are permitted are those that are nearly orthogonal to $[0, 1]$. When the constraint is exactly $c = 1$, the only vectors that are permitted are those that are exactly orthogonal to $[1, 0]$ and have unit norm or less.

This toy example demonstrates graphically that the set of vectors $[x, y]$ admissible under the constraint $||\mathbf{M}||_2 \leq 1$ is the convex hull of the set of vectors $[x, y]$ admissible under the constraint $\mathbf{M}\mathbf{M}^T = \mathbf{I}_2$. Moreover, we see that as $\epsilon \to 0$ the set of vectors $[x, y]$ admissible under the constraint $||\mathbf{M}||_2 \leq 1 + \epsilon$ shrinks towards the set of vectors $[x, y]$ that are orthogonal to the filled-in row of $\mathbf{M}$.

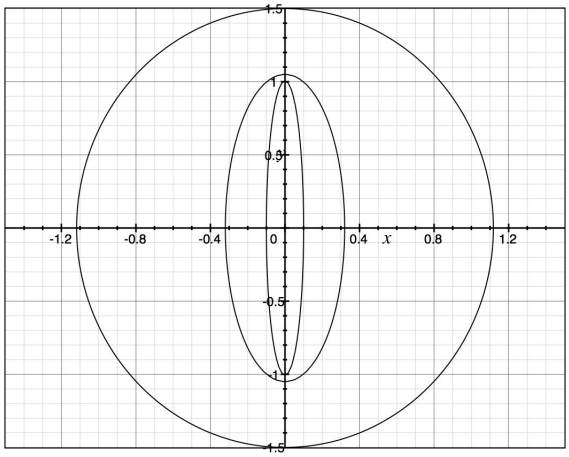


Figure 8: Level sets of the function $\left\| \begin{bmatrix} 1 & 0 \\ x & y \end{bmatrix} \right\|_2$. The outer circle is the 1.1 level set, the middle ellipse is the 1.01 level set, and the inner ellipse is the 1.001 level set.

6.6.2 Optimization

We solve (28) using the proximal gradient method, with the splitting

$$f(\mathbf{M}) = \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 \qquad g(\mathbf{M}) = \mathbf{1}_{||\mathbf{M}||_2 \leq c}$$

The gradient of f is the same as before.

Since g is the indicator function of a set $\mathcal{B} = \{\mathbf{M} : ||\mathbf{M}||_2 \leq c\}$, the proximal operator of g is a projection onto $\mathcal{B}$ — and indeed, the proximal gradient algorithm reduces to projected gradient decent. To project a matrix onto $\mathcal{B}$, the c-ball of the spectral norm, one need only compute the singular value decomposition, and clip all singular values that are greater than c back down to c. That is,

$$\mathbf{prox}_{\mathbf{1}_{\mathcal{B}}}(\mathbf{M}) = \mathbf{U}\,\mathrm{diag}(\max(\boldsymbol{\sigma}, c))\mathbf{V}^T \tag{29}$$

where $\mathbf{M} = \mathbf{U}\,\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^T$ is the SVD of $\mathbf{M}$.

The reason why this is so is that the spectral norm is an orthogonally invariant matrix norm [32, p. 192] whose corresponding gauge function is the vector ∞-norm, so projecting a matrix onto the spectral norm c-ball can be done by projecting its singular values onto the ∞-norm c-ball.

6.7 Simultaneous distinctness and sparsity

To encourage the spatial maps to be both sparse and distinct from one another, we may simultaneously impose a spectral norm constraint and an ℓ_1 penalty:

$$\underset{\mathbf{M}}{\arg\min} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \alpha\,||\mathbf{M}||_1 \quad \text{subject to} \quad ||\mathbf{M}||_2 \leq c \tag{30}$$

where α controls the strength of the sparsity regularization and c controls the strength of the distinctness regularization.

6.7.1 Optimization

The vanilla proximal gradient method cannot handle problems like (30) with multiple nonsmooth terms. An extension proposed in [33] can do so, but we found that in practice a different convex optimization algorithm, the alternating direction method of multipliers (ADMM), solves the problem faster and with less tuning required.

ADMM solves problems of the form:

$$\begin{aligned} \underset{\mathbf{X},\mathbf{Z}}{\arg\min} \quad & f(\mathbf{X}) + g(\mathbf{Z}) \\ \text{subject to} \quad & \mathbf{A}_1\mathbf{X} = \mathbf{A}_2\mathbf{Z} \end{aligned} \tag{31}$$

where f and g are convex functions.

We use the splitting:

$$\begin{aligned} f(\mathbf{X}) &= \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{X}||_F^2 \\ g\left(\begin{bmatrix}\mathbf{Z}_1 \\ \mathbf{Z}_2\end{bmatrix}\right) &= g_1(\mathbf{Z}_1) + g_2(\mathbf{Z}_2) \\ g_1(\mathbf{Z}_1) &= \alpha||\mathbf{Z}_1||_1 \\ g_2(\mathbf{Z}_2) &= \mathbf{1}_{||\mathbf{Z}_2||_2 \leq c} \end{aligned} \tag{32}$$

The derivation of the algorithm is messy and is therefore deferred to the appendix. However, one interesting point of difference from PGM is that ADMM requires the proximal operator of f rather than the gradient. The proximal operator of the squared loss term f is essentially a ridge regression:

$$\begin{aligned} \mathbf{prox}_{\lambda f}(\mathbf{X}) &= \underset{\mathbf{C}}{\arg\min}\, ||\mathbf{Y} - \mathbf{W}\mathbf{C}||_F^2 + \frac{1}{2\lambda}||\mathbf{C} - \mathbf{X}|| \\ &= \left(\mathbf{W}^T\mathbf{W} + \frac{1}{\lambda}\mathbf{I}_k\right)^{-1}\left(\mathbf{W}^T\mathbf{Y} + \frac{1}{\lambda}\mathbf{X}\right) \end{aligned} \tag{33}$$

This fact is derived in Appendix C.

6.8 A "wedge" penalty for distinctness

An alternative way to promote distinctness in the spatial maps is to penalize the ℓ_1 norm of the the inner products between the rows of $\mathbf{M}$. The ℓ_1 penalty will drive these inner products to zero, causing the spatial maps to be pairwise orthogonal.

Unfortunately, the penalty

$$\Omega(\mathbf{M}) = \alpha \sum_{i \neq j} |\mathbf{m}_i^T \mathbf{m}_j| \tag{34}$$

is not convex in $\mathbf{M}$. However, it turns out that one can make the penalty convex by adding ridge terms $||\mathbf{m}_i||_2^2$ with sufficiently large weights.

In particular, the function

$$\begin{aligned}\Omega(\mathbf{M}) &= \alpha \sum_{i,j} \mathbf{K}_{ij} \left|\mathbf{m}_i^T \mathbf{m}_j\right| \\ &= \alpha \text{Tr}(\mathbf{K} \left|\mathbf{M}\mathbf{M}^T\right|)\end{aligned} \tag{35}$$

parameterized by some kernel matrix $\mathbf{K} \in \mathbb{R}^{K \times K}$ is convex if the so-called "comparison matrix" $\tilde{\mathbf{K}} \in \mathbb{R}^{K \times K}$ defined by

$$\tilde{K}_{ij} = \begin{cases} K_{ij} & \text{if } i = j \\ -K_{ij} & \text{if } i \neq j \end{cases} \tag{36}$$

is positive semidefinite [42, Theorem 3.1] or [21, Theorem 3.2].

Following [39], we consider kernels where the off-diagonal entries are 1 and the diagonal entries are θ:

$$\mathbf{K}_\theta = \begin{bmatrix} \theta & 1 & 1 \\ 1 & \ddots & 1 \\ 1 & 1 & \theta \end{bmatrix}$$

which yields the penalty:

$$\Omega(\mathbf{M}) = \alpha \left[2 \sum_{i \neq j} |\mathbf{m}_i^T \mathbf{m}_j| + \theta \sum_i ||\mathbf{m}_i||_2^2 \right] \tag{37}$$

We call (37) the *wedge* penalty, since its effect is to place a "wedge" between the rows of $\mathbf{M}$, preventing them from being too alike.

The wedge penalty is non-convex when $\theta < K$ (K is the number of rows of $\mathbf{M}$), convex when $\theta \geq K$, and strictly convex when $\theta > K$.

However, this convexity comes at a cost: when θ is large, the distinct-ifying effect of the wedge penalty is diminished. As we will see below, it is often unfortunately the case that when θ is large enough that Ω is convex, the wedge does not encourage distinctness at all. We will still try to optimize a non-convex Ω, but with no guarantees that the stationary point we reach is a global minimum.

Learning the spatial maps under the wedge penalty entails solving the following optimization problem:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \alpha \left[2 \sum_{i \neq j} |\mathbf{m}_i^T \mathbf{m}_j| + \theta \sum_i ||\mathbf{m}_i||_2^2 \right] \tag{38}$$

6.8.1 Intuition

To provide some intuition for the wedge penalty, we take a page from [39] and plot the unit level sets of the function

$$\Omega_\theta \left(\begin{bmatrix} x & y \\ y & z \end{bmatrix} \right)$$

as a function of (x, y, z). We plot the level sets of Ω_θ for three settings of θ: a setting for which the penalty is nonconvex ($\theta = \frac{1}{2}$), a setting for which the penalty is convex but not strictly convex ($\theta = 1$), and a setting for which the penalty is strictly convex ($\theta = \frac{3}{2}$).

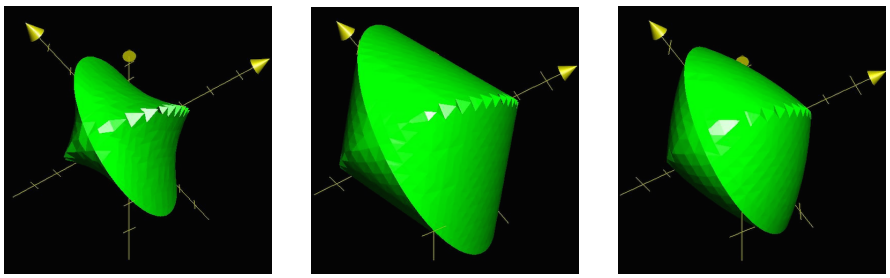


Figure 9: Level sets of the wedge penalty for nonconvex (left), convex (middle) and strictly convex (right) settings of θ, the strength of the ridge penalty on the spatial maps.

The "edges" of the level surface of the wedge penalty are two circles, which correspond to the points where the two vectors (x, y) and (y, z) are exactly orthonormal: one circle is at $y = 0$, and the other circle is at $x = -z$.

When penalty is non-convex, the level surface is bowed inwards from the two circles; when the penalty is convex but not strictly convex, the surface is flat between the two circles; and when the penalty is strictly convex, the surface is bowed outward. Intuitively, it makes sense that a function being minimized over this level set would be more likely to take on its minimal value at one of the orthonormal "edges" when the surface is bowed inwards than when the surface is bowed outwards. Unfortunately, these are precisely the settings of θ for which the wedge penalty is not convex.

6.8.2 Related work

[42] used this penalty as a regularizer in a hierarchical SVM, to encourage the hyperplane at each juncture to be nearly orthogonal to the hyperplane of the parent juncture. [39] found that choosing $\mathbf{K}$ so as to make the penalty non-convex is more effective at promoting orthogonality than choosing a $\mathbf{K}$ that makes the penalty convex. [21] analyzed a larger class of penalty functions intended to promote various pairwise relationships between vectors, including but not limited to orthogonality. [42] optimized the penalty using regularized dual averaging, [39] did so using the subgradient method, and [21] proposed a mirror-prox algorithm for the special case where the loss function can be formulated as a variational problem.

6.8.3 Optimization

The wedge penalty is nonsmooth, so we cannot compute a gradient of (37). Therefore, following [39], we optimize (38) using the subgradient method.

A vector $\mathbf{g} \in \mathbb{R}^n$ is called a *subgradient* of a function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $\mathbf{x}_0 \in \mathbb{R}^n$ if f lies entirely above its linear approximation around $\mathbf{x}_0$ with slope $\mathbf{g}$. That is, $\mathbf{g}$ is a subgradient if, for all $\mathbf{x} \in \mathbb{R}^n$,

$$f(\mathbf{x}) \geq \mathbf{g}^T(\mathbf{x} - \mathbf{x}_0) + f(\mathbf{x}_0)$$

The set of all subgradients of f at $\mathbf{x}$ is called the *subdifferential* of f at $\mathbf{x}$ and is denoted $\partial f(\mathbf{x})$.

The subgradient method is a generalization of gradient descent to non-smooth objective functions [35, 4]. The idea is to take steps in the opposite direction of a subgradient:

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t \, \mathbf{g}_t \quad \text{where} \quad \mathbf{g}_t \in \partial f(\mathbf{x}_t) \tag{39}$$

When f is convex, the subgradient method is guaranteed to eventually get arbitrarily close to a global minimizer. In our case, when the wedge penalty is not convex, no such guarantee exists; nevertheless, the subgradient method appears to work well in practice.

A subgradient $\mathbf{G}_t$ of Ω at $\mathbf{M}_t$ is given by:

$$\mathbf{G}_t = \begin{bmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_K \end{bmatrix} \qquad \mathbf{g}_k = 2 \left[\sum_{i \neq k} \text{sign}(\mathbf{m}_k^T \mathbf{m}_i) \; \mathbf{m}_i + \theta \, \mathbf{m}_i \right] \tag{40}$$

Hence, each iteration of the subgradient method for minimizing (38) is:

$$\mathbf{M}_{t+1} = \mathbf{M}_t - \eta_t \left(\mathbf{W}^T \mathbf{W} \mathbf{M}_t - \mathbf{W}^T \mathbf{Y} + \mathbf{G}_t \right) \tag{41}$$

We use step sizes that decrease as

$$\eta_t = \frac{\eta_0}{\sqrt{t}} \tag{42}$$

We also tried both a constant step size and AdaGrad [14], an algorithm that adaptively chooses a different step size for each parameter; we found that the step size schedule in (42) was the most effective of the three alternatives.

Choosing the initial step size η_0 proved to be tricky: if η_0 is too large, the optimization will diverge; if η_0 is too small, the procedure will take too long. Therefore, each time we solve (38), we start with η_0 high and run the subgradient method. If the algorithm diverges before reaching $T = 5,000$ iterations, we decrease the initial step size and start over. If the algorithm runs for T iterations without causing the objective function to increase, we return its final iterate $\mathbf{M}_T$.

7 Qualitative Comparison of Penalties

We have described constraints and penalties on the spatial maps that promote distinctness, sparsity, and smoothness, and we have given algorithms for learning the spatial maps under each of these penalties. In this section, we will qualitatively compare and contrast the effects of each of these penalties by visualizing, side by side, the spatial maps they yield under the same task.

In what follows, we initialize $\mathbf{W}$ by running a single step of alternating minimization on the Twilight Zone dataset; then we try to learn subject 1's spatial maps $\mathbf{M}$ based on the shared timecourses $\mathbf{W}$ and subject 1's data $\mathbf{Y}$, subject to various penalties on the spatial maps.

Specifically, we solve:

$$\arg\min_{\mathbf{M}} \; \frac{1}{2} ||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \Omega(\mathbf{M})$$

where $\Omega(\mathbf{M})$ is the penalty.

We compare the following penalties:

name	$\Omega(\mathbf{M})$	intended effect	comments
least squares	0	none	
ℓ_1	$\alpha\|\|\mathbf{M}\|\|_1$	sparsity	
orthogonal	$\mathbf{1}_{\mathbf{M}\mathbf{M}^T=c^2\mathbf{I}}$	distinctness	non-convex
spectral	$\mathbf{1}_{\|\|\mathbf{M}\|\|\leq c}$	distinctness	
spectral + ℓ_1	$\mathbf{1}_{\|\|\mathbf{M}\|\|\leq c} + \alpha\|\|\mathbf{M}\|\|_1$	distinctness, sparsity	
wedge	$\alpha\left[\sum_{k_1\neq k_2} \lvert\mathbf{m}_{k_1}^T\mathbf{m}_{k_2}\rvert + \theta\sum_k \|\|\mathbf{m}_k\|\|^2\right]$	distinctness	convex iff $\theta \geq K-1$
smooth + ℓ_1	$\alpha\|\|\mathbf{M}\|\|_1 + \beta\|\|\mathbf{M}\|\|_F^2 + \gamma\sum_k \mathbf{m}_k^T\mathbf{L}\,\mathbf{m}_k$	smoothness, sparsity	$\mathbf{L}$ is the graph laplacian

Figure 10 shows the spatial maps learned under various penalties. Notice the following facts:

1. Maps learned under the orthogonal constraint (row 2) have more "definition" (in the "boxflex commercial" sense of the word) than maps learned with no penalties (row 1).
2. A spectral norm bound of $c = 1.0$ is apparently small enough that the maps learned under this constraint (row 4) are almost identical to those learned under the orthogonal constraint (row 2).
3. Spatial maps learned under an ℓ_1 penalty (row 3) and a spectral + ℓ_1 penalty (row 5) are both sparse, though the latter maps are more distinct (for example, compare columns 4 and 6 under both penalties).
4. Maps learned under the convex setting of the wedge penalty (row 6) have better definition than the unconstrained maps (row 1), but maps learned under the *non-convex* setting of the wedge penalty (row 7) are better still.
5. Maps learned under the smooth + ℓ_1 penalty (row 8) are smooth and sparse.

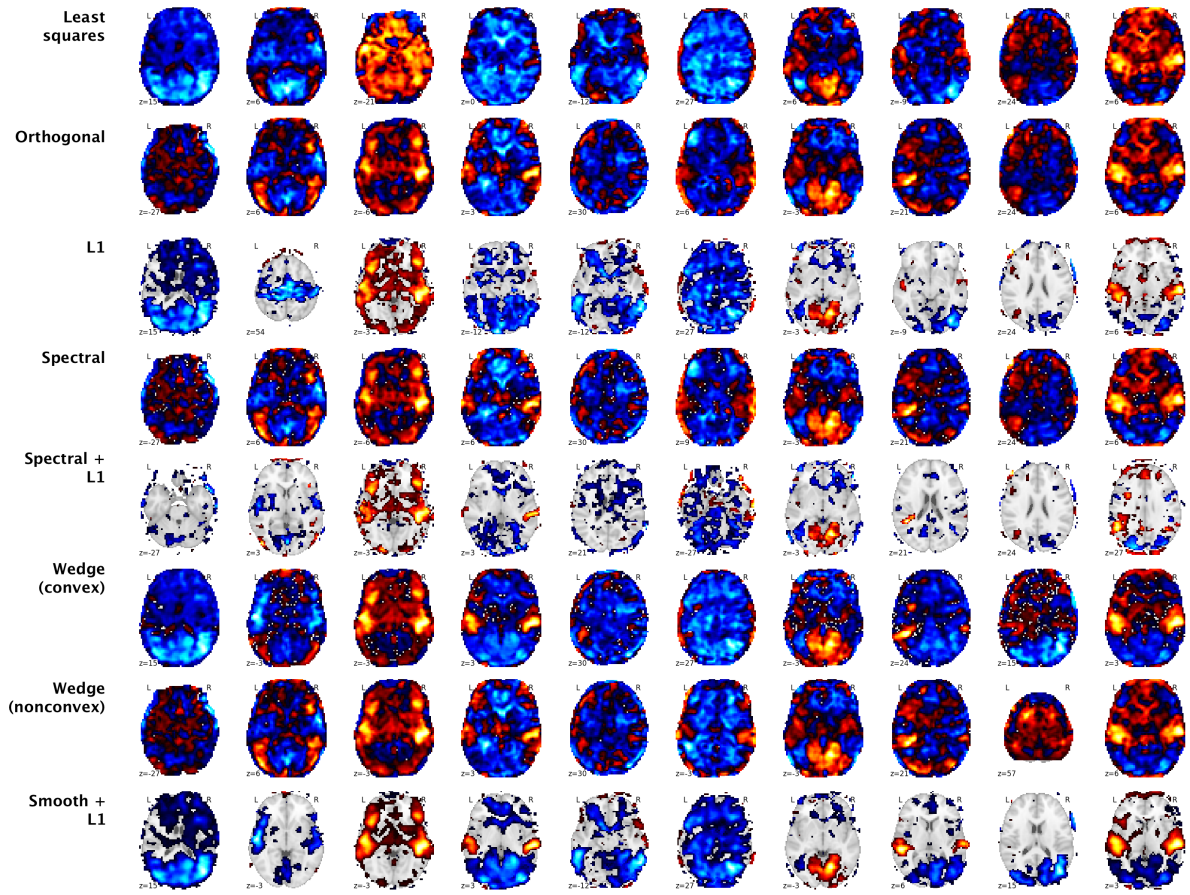


Figure 10: $K = 10$ spatial maps learned under various penalties. Each column k shows the spatial map for component k (represented with a single horizontal slice) learned under different penalties (rows).

7.1 Distinctness

In this section, we evaluate the distinct-ifying effect of each of the penalties. That is, we ask: how forcefully does each penalty encourage the spatial maps to be distinct from one another?

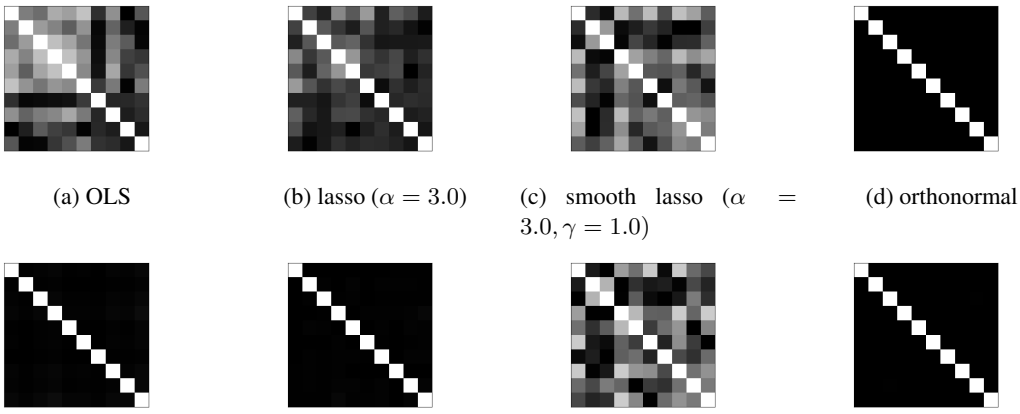
A measure of the similarity between two spatial maps $\mathbf{m}_1 \in \mathbb{R}^V$ and $\mathbf{m}_2 \in \mathbb{R}^V$ is *absolute cosine similarity*: the magnitude of the cosine of the angle between the two vectors:

$$|\cos(\mathbf{m}_1, \mathbf{m}_2)| = \frac{|\langle \mathbf{m}_1, \mathbf{m}_2 \rangle|}{||\mathbf{m}_1|| ||\mathbf{m}_2||} \tag{43}$$

This quantity is one when the vectors are parallel or antiparallel, and zero when they are orthogonal.

We measure the overall "distinctness" of a set of spatial maps $\mathbf{m}_1$ through $\mathbf{m}_K$ with a *similarity matrix*: a $K \times K$ matrix containing the absolute cosine similarity between each pair of maps. The diagonal elements of this matrix are always one, because each spatial map is parallel to itself. The off-diagonal elements of this matrix are near one when the spatial maps are similar, and near zero when they are distinct.

Figure 11 shows similarity matrices for spatial maps learned under different penalties from the same $\mathbf{W}$ and $\mathbf{Y}$. One can see that the orthonormal constraint, the spectral constraint with $c = 1$, and the wedge constraint with $\theta = 0$ all induce near-orthogonality of the spatial maps. The lasso, too, makes the spatial maps more distinct than plain least-squares, presumably because many of the voxels are zero-ed out.

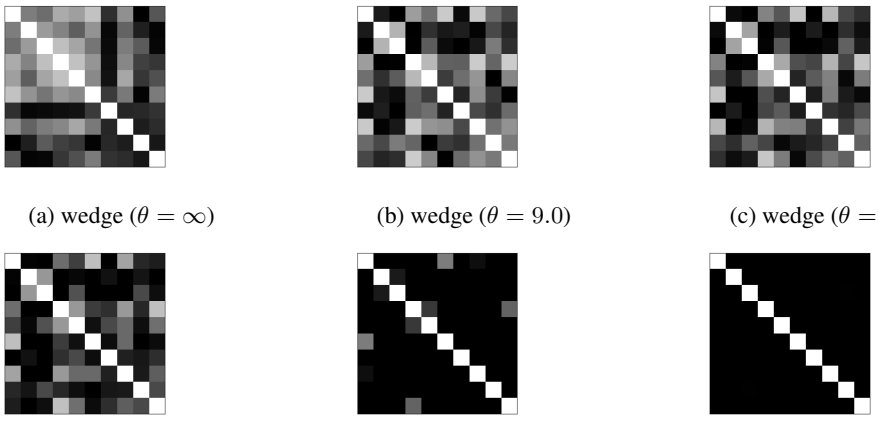


(a) OLS (b) lasso ($\alpha = 3.0$) (c) smooth lasso ($\alpha = 3.0, \gamma = 1.0$) (d) orthonormal

(e) spectral ($c = 1.0$) (f) spectral + ℓ_1 ($c = 1.0$, $\alpha = 3.0$) (g) wedge ($\theta = 9.0$) (h) wedge ($\theta = 0.0$)

Figure 11: Similarity matrix between the spatial maps $(K = 10)$ learned by various penalties. The similarity matrix is computed as in (43).

Both the spectral constraint and the wedge penalty have a tuning parameter that controls the degree to which distinctness is enforced. Figure 12 shows the similarity matrix under the wedge penalty as θ, the strength of the ℓ_2 penalty on the maps, is decreased. The lower the θ, the more the maps are forced to be orthogonal.

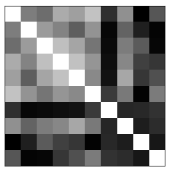


(a) wedge ($\theta = \infty$) (b) wedge ($\theta = 9.0$) (c) wedge ($\theta = 5.0$)

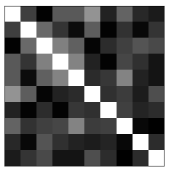
(d) wedge ($\theta = 3.0$) (e) wedge ($\theta = 1.0$) (f) wedge ($\theta = 0.0$)

Figure 12: Similarity matrix between the spatial maps $(K = 10)$ learned by the wedge penalty $(\alpha = 1)$, as θ, the weight of the ℓ_2 penalty on the spatial maps, is decreased from 9.0 to 0.0. At $\theta = 9.0$, the wedge penalty is convex; for lower values, the penalty is not convex.

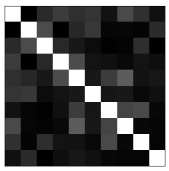
Similarly, figure 13 shows the similarity matrix for maps learned under the spectral norm constraint as the spectral norm bound c is decreased. When c is greater than the *largest* singular value of the OLS solution $\sigma_1(\mathbf{M}_{\text{OLS}})$, the constraint is not "binding," and the solution is the same as the OLS solution. Conversely, when c is equal to the *smallest* singular value of the OLS solution $\sigma_{10}(\mathbf{M}_{\text{OLS}})$, the resulting spatial maps are exactly orthogonal, and reducing c further will just scale the solution down by a constant factor.



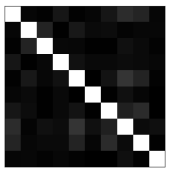
(a) spectral ($c = \infty$)



(b) spectral ($c = \sigma_1$)

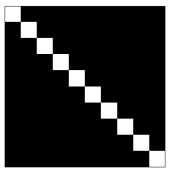


(c) spectral ($c = \sigma_3$)



(d) spectral ($c = \sigma_5$)

(e) spectral ($c = \sigma_7$)



(f) spectral ($c = \sigma_{10}$)

Figure 13: Similarity matrix between the spatial maps ($K = 10$) learned by the spectral constraint as c, the bound on the spectral norm of the spatial maps matrix, is decreased from σ_1, the largest singular value of the least-squares solution, to σ_{10}, the smallest.

8 Experiments

We would like to know which SRM penalties yield spatial maps that provide an informative basis for other fMRI data collected from the same subjects. To that end, we evaluate the different penalties using both a decoding experiment and a matching experiment. In the decoding experiment, we ask whether subject-specific spatial maps learned from a movie-viewing dataset provide a good basis for performing a "mind-reading" feat on those same subjects. In the matching experiment, we ask whether subject-specific spatial maps learned from the first half of a movie provide a good basis for matching a randomly chosen time segment from the second half of the movie between different subjects.

We benchmark five different SRM algorithms:

1. Unconstrained SRM, which does not constrain or penalize the spatial maps. Unconstrained SRM does not have any tuning parameters.
2. Orthogonal SRM, which constrains the spatial maps to be distinct via an orthogonality constraint. We fixed the orthogonality constraint at $c = 1$, i.e. $\mathbf{M}\mathbf{M}^T = \mathbf{I}$, so orthogonal SRM does not have any tuning parameters.
3. Smooth SRM, which encourages the spatial maps to be smooth and sparse via the "smooth lasso" described in section 6.4. Smooth SRM has three tuning parameters: α, the strength of ℓ_1 regularization, β, the strength of ℓ_2 regularization, and γ, the strength of smoothness regularization.

   In all our experiments below, cross-validation always chose $\gamma = 0$, implying that the smoothness penalty is actually detrimental to performance. Therefore, we refer to smooth SRM below as "sparse SRM," as it in practice just involves an elastic net penalty.
4. (Spectral + ℓ_1) SRM, which encourages the spatial maps to be sparse and distinct via a spectral norm constraint combined with an ℓ_1 penalty, as described in section 6.7. We fix the spectral norm constraint at $c = 1$, so (spectral + ℓ_1) SRM has just one tuning parameter: α, the strength of ℓ_1 regularization.
5. Wedge SRM, which encourages the spatial maps to be distinct via a wedge penalty, as described in section 6.8. We fix $\alpha = 1.0$, so wedge SRM has just one tuning parameter: θ, the strength of the ℓ_2 regularization.

We also benchmarked a "hack" that might serve as an alternative way to simultaneously enforce distinctness and sparsity: run orthogonal SRM to completion and then hard-threshold the resulting spatial maps. We call this method "(orthogonal + threshold) SRM." It has one tuning parameter: the threshold.

In this thesis we only report results from running the SRM algorithms with $K = 10$ components. However, we observed similar results from runs with $K = 20$ components.

8.1 Matching Experiment

In the matching experiment, we ask if spatial maps learned using the first half of a movie provide an informative basis for "matching" timecourses in the second half of the movie between subjects, under the assumption that the synchronized stimulus induces a similar neural response from subject to subject.

8.1.1 Datasets

We use three movie-viewing datasets:

1. **Twilight Zone**: 24 subjects watched a 25-minute black-and-white "Twilight Zone" episode, "The Lateness of the Hour" [10]. The recording lasts 1028 TRs.[1]
2. **Sherlock**: 16 subjects watched a 50-minute segment of the BBC's "Sherlock," "A Study in Pink" [11]. The recording lasts 1976 TRs.
3. **Raiders**: 10 subjects watched "Raiders of the Lost Ark." The recording lasts 1106 TRs.

For the first two datasets, the following preprocessing steps were performed in FSL: slice-time correction, motion correction, linear detrending, high -pass filtering (140s cutoff), and coregistration and affine transformation to the MNI template brain. Finally, each voxel's time-series was z-scored — that is, centered and normalized to have zero mean and unit norm.

Preprocessing for the Raiders dataset is described below.

8.1.2 Experimental Procedure

We use the following procedure to evaluate a given SRM algorithm $\mathcal{A}$ with parameter θ:

Let $\mathbf{Y}_i \in \mathbb{R}^{N \times V_i}$ be an fMRI recording of a synchronized stimulus from subject i, with N the number of TRs and V_i the number of voxels.

Let $\mathbf{Y}_i^1$ be the first $N/2$ TRs from $\mathbf{Y}_i$ and let $\mathbf{Y}_i^2$ be the last $N/2$ TRs, so that $\mathbf{Y}_i = \begin{bmatrix} \mathbf{Y}_i^1 \\ \mathbf{Y}_i^2 \end{bmatrix}$.

Matching procedure to evaluate $\mathcal{A}_\theta$

1. Run $\mathcal{A}_\theta$ on the first half of the movie $\{\mathbf{Y}_i^1\}_{i=1}^S$. This will return a set of spatial maps for each subject $\{\mathbf{M}_i^1\}_{i=1}^S$ and a shared set of timecourses $\mathbf{W}^1$.

$$\mathbf{W}^1, \{\mathbf{M}_i^1\}_{i=1}^S \leftarrow \mathcal{A}\left(\{\mathbf{Y}_i^1\}_{i=1}^S\right)$$

2. Pick M time segments, each of length Δt, from the second half of the movie. Call these $\{(t_m, t_m + \Delta t)\}_{m=1}^M$.

   We used $\Delta t = 10$, corresponding to a time segment length of 15 seconds for the Twilight Zone dataset.

3. Do the following for each subject $i \in [S]$:

   Using subject maps $\mathbf{M}_i^1$ learned from the first half of the movie and data $\mathbf{Y}_i^2$ from the second half of the movie, separately solve for both (a) the timecourses of subject i, $\mathbf{W}_{i,\text{same}}^2$; and (b) the shared

[1] The first 26 TRs were actually recorded during eye-tracking experiment rather than the actual TV episode viewing. These TRs were erroneously fed to the algorithm.

timecourses of all other subjects, $\mathbf{W}^2_{i,\text{other}}$.

$$\mathbf{W}^2_{i,\text{same}} = \arg\min_{\mathbf{W}} ||\mathbf{Y}^2_i - \mathbf{W}\mathbf{M}^1_i||^2_F \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1$$

$$\mathbf{W}^2_{i,\text{other}} = \arg\min_{\mathbf{W}} \sum_{j \neq i} ||\mathbf{Y}^2_j - \mathbf{W}\mathbf{M}^1_j||^2_F \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1$$

For each of the M time segments $m = 1 \ldots M$:

(a) Let $\Delta\mathbf{w} \in \mathbb{R}^{\Delta t \times K}$ be the time segment $(t_m, t_m + \Delta t)$ in $\mathbf{W}^2_{i,\text{same}}$.

(b) Try to guess t_m based on $\Delta\mathbf{w}$ and $\mathbf{W}^2_{i,\text{other}}$:

i. Consider each time segment of length Δt in $\mathbf{W}^2_{i,\text{other}}$ that does not overlap with $(t_m, t_m + \Delta t)$, and rank these in order of Frobenius norm difference from $\Delta\mathbf{w}$.

ii. If $(t_m, t_m + \Delta t)$ is in the top `THRESHOLD` (= 20) of the ranked time segments, report "success." Otherwise, report "failure."

The fraction of the M time segments that were "successfully" identified is the accuracy of the matching task for subject i for the second movie half.

4. Repeat steps 1-3 with the roles of the halves reversed. That is, learn spatial maps using data from the second half of the movie, and evaluate them on data from the first half of the movie.

5. Return the mean matching accuracy, where the mean is taken over all subjects and both movie halves.

To make a fair comparison between different SRM algorithms, we need a way to set the parameter θ of each algorithm without "double-dipping" into the test data that we evaluate the algorithm on. To that end, we split the subjects into two folds and employ two-fold cross validation, using one fold to tune the parameter and the other fold to evaluate the algorithm.

For a given SRM algorithm $\mathcal{A}$, let Θ be a grid of parameter values to be considered.

To "score" an SRM algorithm $\mathcal{A}$:

1. Split the subjects into two folds S_1 and S_2.

2. For each $\theta \in \Theta$, run the above matching procedure on subjects in S_1. Let $\hat{\theta}$ be the parameter setting that yields the highest mean matching accuracy. Run the matching procedure on subjects in S_2 using parameter setting $\hat{\theta}$, and report the mean matching accuracy.

3. Repeat step 2, but with the roles of S_1 and S_2 reversed.

4. Return the average, over both subjects-folds, of the mean matching accuracy.

8.1.3 Results

Figure 14 shows the mean accuracy of each SRM algorithm on the matching task for both the Sherlock and Twilight Zone datasets. The main results:

1. Sparse SRM and wedge SRM just barely outperform unregularized SRM.

2. Orthogonal SRM outperforms sparse SRM, wedge SRM, and unregularized SRM.
3. (Spectral + ℓ_1) SRM and (orthogonal + threshold) SRM outperform all of the other algorithms by a considerable margin. It is unclear which (if either) of the two methods to jointly induce sparsity and distinctness is better in general: (spectral + ℓ_1) performed better on the Twilight Zone dataset, while (orthogonal + threshold) performed better on the Sherlock and Raiders datasets.

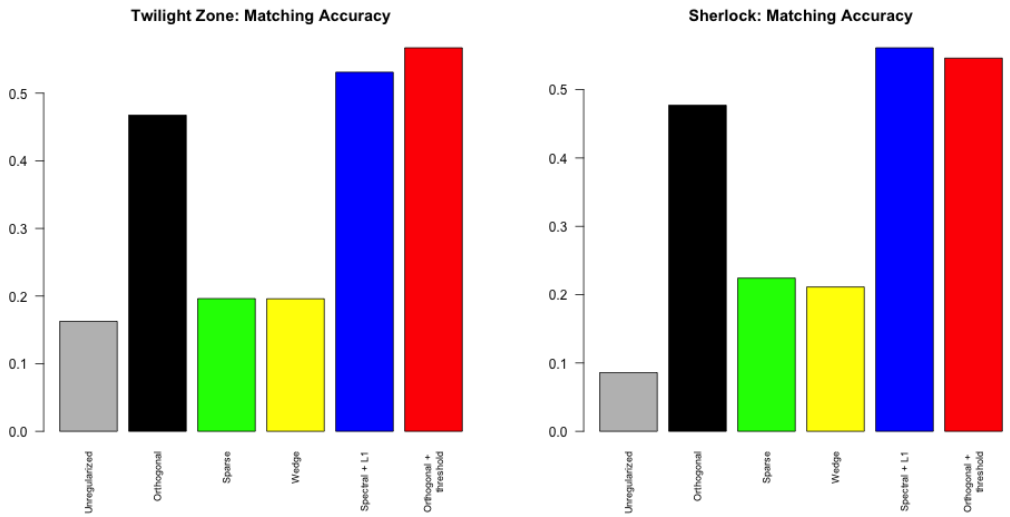


Figure 14: Matching accuracy for various SRM algorithms on the Twilight Zone, Sherlock, and Raiders movie-viewing datasets. To set each algorithm's parameter, subjects were divided into two folds. One fold was used to tune the parameter, and the other was used to evaluate the algorithm. The matching accuracy reported is the average over both evaluation folds.

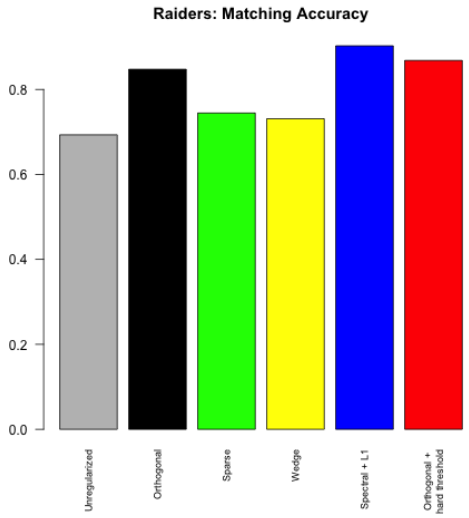


Figure 15 shows how the matching accuracy of (spectral + ℓ_1) SRM varies with the strength of the sparsity regularization. The horizontal solid line in the figure is the matching accuracy of orthogonal SRM. At low

levels of sparsity, (spectral + ℓ_1) SRM is beaten by orthogonal SRM, probably because the latter algorithm enforces a stricter distinctness constraint on the spatial maps. However, as the sparsity knob is turned up, (spectral + ℓ_1) SRM soon overtakes its un-sparse competitor. Nevertheless, it is possible to have too much of a good thing — when the sparsity level is cranked up too high, performance starts to drop.

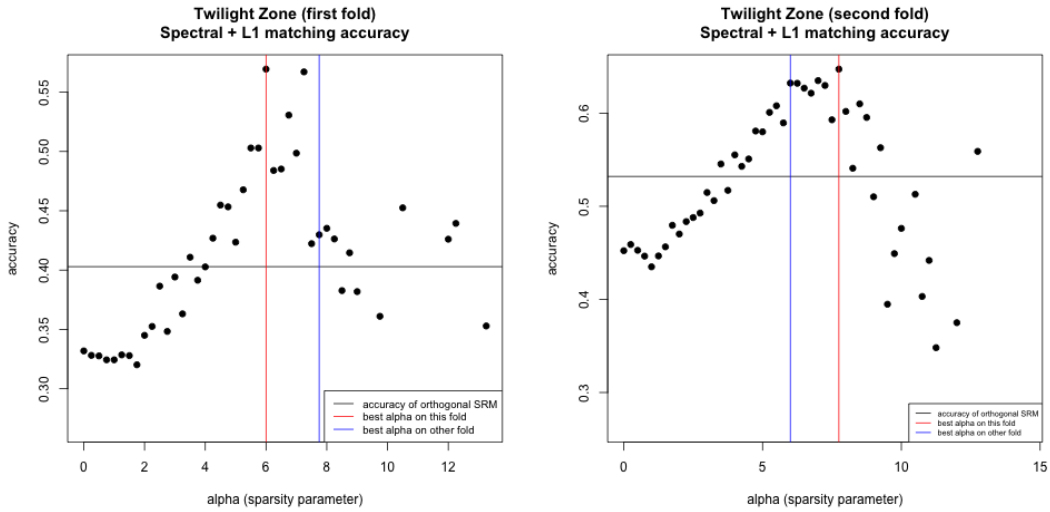
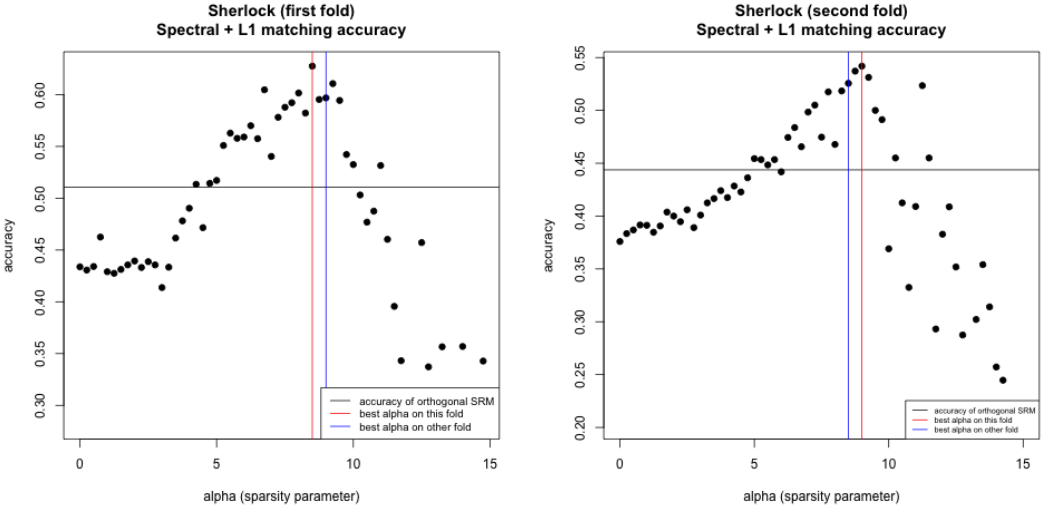


Figure 15: Matching accuracy of spectral + ℓ_1 SRM on the Twilight Zone dataset (top) and the Sherlock datsaet (bottom) as a function of α, the strength of sparsity regularization. The black horizontal line is the matching accuracy of orthogonal SRM. Accuracy first rises with α, then falls: there is such a thing as too much sparsity. The red vertical line is the α that maximizes matching accuracy on this fold; the blue vertical line is the α that maximizes matching accuracy on the other fold, and hence the α that was used to evaluate the performance of the method on this fold during cross-validation.



In order to verify that we ran each SRM model for a sufficient number of iterations of alternating minimization, figure 16 plots the learning curve for several SRM models over 50 iterations of alternating minimization. The plots show that 50 iterations is sufficient for convergence on our datasets.

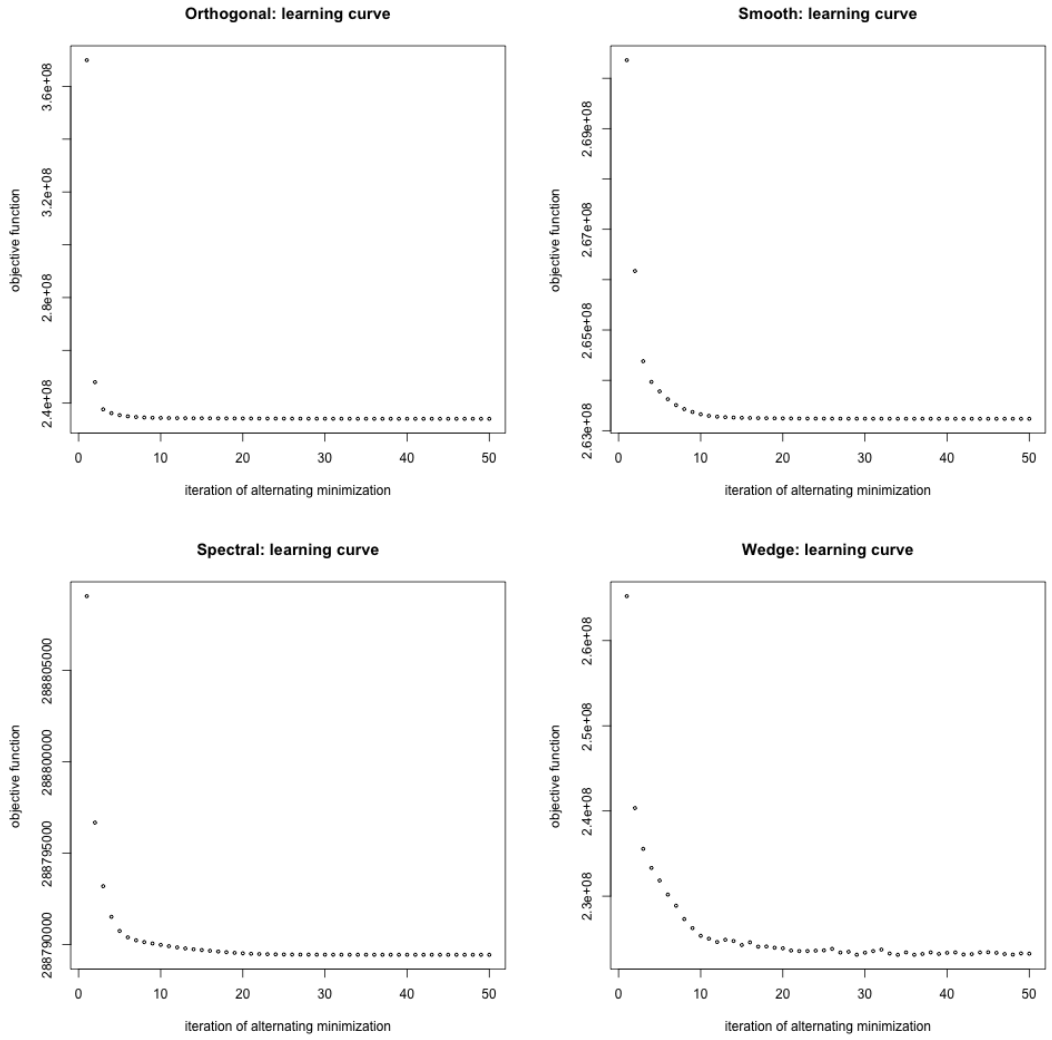


Figure 16: Plot of the overall SRM objective function (4) through 50 iterations of alternating minimization.

8.2 Decoding Experiment

In the decoding experiment, we ask if spatial maps learned from a synchronized stimulus provide a good basis for performing a neural decoding task over the same set of subjects.

8.2.1 Dataset

Ten subjects watched the movie "Raiders of the Lost Ark" in the fMRI scanner, and also viewed a series of images in an image category perception experiment [16].

The movie was presented in two scanner sessions, each lasting approximately 55 minutes, with a short break in between.

The image category perception experiment occurred in a third scanner session. Each subject viewed a series of images from seven categories: male faces, female faces, monkey faces, dog faces, shoes, chairs, and houses. The experiment involved eight *runs* of images, and each run consisted of seven *blocks*, one for each category. In each block, the subject viewed 16 grayscale images from that category. Data from the 11th to the 26th TR of each category block were averaged to represent the response pattern during that category block for later analysis. Thus each subject yielded a total of 56 "examples," eight from each of the seven image categories, on which to perform a neural decoding experiment.

Only data from the 3,000 or so voxels from each subject's ventral temporal (VT) cortex were analyzed. A researcher demarcated each subject's VT cortex by hand.

The following preprocessing steps were performed using the AFNI software package: slice time correction, despiking, low- and high-pass filtering (movie data only), linear and quadratic detrending (image category data only), and spatial smoothing with a 4mm FWHM Gaussian blur. For each subject, data from all three sessions (both movie sessions and the image category recognition experiment) were spatially aligned. Data from each of the three sessions were mean-centered and unit-normalized (z-scored) separately.

8.2.2 Experimental Procedure

We use the following procedure to evaluate a given SRM algorithm $\mathcal{A}$ with parameter θ:

Let $\mathbf{Y}_i \in \mathbb{R}^{N \times V_i}$ be the movie data from the i-th subject, where N is the number of TRs in the movie and V_i is the number of voxels.

Let $\mathbf{X}_i \in \mathbb{R}^{M \times V_i}$ be a matrix with the decoding experiment data from the i-th subject, where M is the number of examples and V_i is the number of voxels.

Let $\mathbf{z}_i \in \{1 \dots C\}^M$ be the class labels for the decoding experiment data, where C is the number of classes and M is the number of examples.

Decoding procedure to evaluate $\mathcal{A}_\theta$

1. Run $\mathcal{A}_\theta$ on the movie $\{\mathbf{Y}_i\}_{i=1}^S$. This will return a set of spatial maps for each subject $\{\mathbf{M}_i\}_{i=1}^S$ and a shared set of timecourses $\tilde{\mathbf{W}}$.

$$\tilde{\mathbf{W}}, \{\mathbf{M}_i\}_{i=1}^S \leftarrow \mathcal{A}_\theta \left(\{\mathbf{Y}_i\}_{i=1}^S\right)$$

2. For each subject i: use the spatial maps learned from the movie $\mathbf{M}_i$ to represent the decoding experiment data in the common shared space as $\mathbf{W}_i \in \mathbb{R}^{M \times K}$:

$$\mathbf{W}_i = \arg\min_{\mathbf{W}} ||\mathbf{X}_i - \mathbf{W}\mathbf{M}_i||_F^2 \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1$$

The n-th row of $\mathbf{W}_i$ is a K-dimensional representation of the state of subject i's brain during the presentation of the n-th example (which has class z_{in}).

3. Do the following for each subject $i \in [S]$:

   Set ϕ, the hyperparameter of a classification algorithm, using a within-subject decoding task on $\mathbf{W}_i$. (For the classification algorithm, we used a support vector machine with an RBF kernel implemented in `scikit-learn` as `sklearn.svm.svc`; its hyperparameter is the level of ℓ_2 regularization.)

   (a) Let Φ be a range of hyperparameter values to be considered. For each $\phi \in \Phi$, evaluate the hyperparameter choice of ϕ in the following way:

   Repeat the following T times:

   i. Split the data $\mathbf{W}_i$ and labels $\mathbf{z}_i$ into a testing set, containing one randomly chosen example from each class, and a training set, containing everything else.

   ii. Train the support vector machine on the training data, and compute its classification accuracy on the test data.

   Return the mean classification accuracy over all T train/test splits.

   (b) Let $\hat{\phi}$ be the $\phi \in \Phi$ with the highest classification accuracy.

   Run a between-subjects decoding task on $\{\mathbf{W}_i\}_{i \neq j}$ with classification algorithm hyperparameter set to $\hat{\phi}$ and report the accuracy:

   (a) For each subject $j \neq i$, train an SVM on data and labels $(\mathbf{W}_\ell, \mathbf{z}_\ell)$ from all $\ell \in [S] \setminus i, j$ and evaluate its classification accuracy on test data and labels $(\mathbf{W}_j, \mathbf{z}_j)$.

   Compute the average classification accuracy over all test subjects j.

   Return the mean classification accuracy over all parameter-tuning subjects i.

As in the matching experiment, in order to get around the problem of choosing a parameter θ for each SRM algorithm $\mathcal{A}$, we split the subjects into two folds and employ two-fold cross validation, using one fold to set $\hat{\theta}$ and the other fold to evaluate $\mathcal{A}_{\hat{\theta}}$.

8.2.3 Results

Figure 17 shows the average decoding accuracy for each of the six SRM algorithms. The main result is that explicit regularization for distinctness improves decoding accuracy (compare the black bar to the gray bar), but regularizing for sparsity *and* distinctness together does not perform any better than just regularizing for distinctness. (Spectral + ℓ_1) SRM and (orthogonal + threshold) SRM performed equally well, but no better than orthogonal SRM.

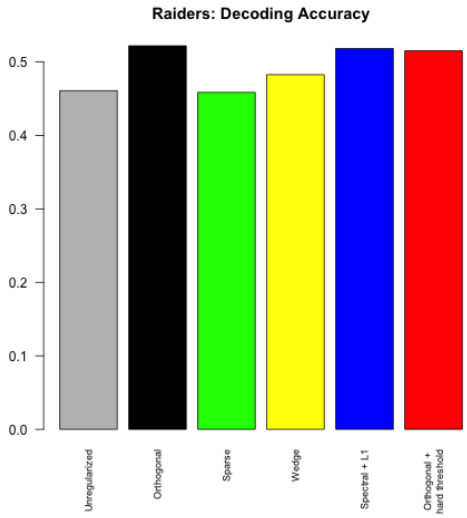


Figure 17: Accuracy with which a support vector machine trained on fMRI data "translated" into the shared space learned from movie-viewing data could predict which of seven image categories a subject was looking at. To set each SRM algorithm's parameter, subjects were divided into two folds. One fold was used to tune the parameter, and the other was used to evaluate the algorithm. The decoding accuracy reported is the average over both evaluation folds.

9 Conclusion

In this work we studied several penalty functions for fMRI data analysis, each of which encodes a different assumption about the topography of brain systems:

1. The ℓ_1 penalty assumes that each brain system is *sparse* — that is, present in only a few voxels.
2. The "smooth lasso" penalty assumes that each brain system is *smooth* — that is, concentrated in voxels that are nearby each other.
3. The orthogonal penalty, the spectral norm constraint, and the "wedge" penalty each assume that the brain systems are *distinct*, or non-overlapping in physical space.

To simultaneously encourage *both* distinctness *and* sparsity, we proposed combining an ℓ_1 penalty with a spectral norm constraint, and we derived an alternating direction method of multipliers algorithm that can optimize this joint "spectral + ℓ_1" penalty.

To our knowledge, neither the spectral norm constraint nor the wedge penalty have been previously explored in the fMRI literature. In fact, the spectral norm constraint does not appear to have been studied thoroughly in the statistics / machine learning literature either.

We used these penalties as regularization on the subject-specific spatial maps in a linear "shared response model." A linear SRM assumes that multiple subjects viewing a "synchronized stimulus" such as a movie experience the exact same timecourse of activations in some lower-dimensional shared space, but that each dimension in this shared space is associated with a different spatial map in each subject. "Solving" or "fitting" an SRM involves finding the shared timecourses and subject-specific spatial maps that maximize data fit while

also respecting an optional penalty term that "punishes" undesirable properties in the spatial maps (whatever those may be). After fitting an SRM, one can use the learned spatial maps to "translate" additional fMRI data collected from the same set of subjects into a low-dimensional shared space in which the data can be meaningfully pooled across subjects — say, to train a classifier in a neural decoding experiment.

We used both a timecourse matching experiment and a neural decoding experiment to evaluate the degree to which the spatial maps learned under each penalty serve as an informative basis for additional fMRI data collected from the same subjects. In the timecourse matching experiment, sparsity and distinctness each proved beneficial, while smoothness proved detrimental. The SRMs that worked best were (spectral + ℓ_1) SRM and (orthogonal + threshold) SRM, which encouraged both sparsity and distinctness. In the neural decoding experiment, on the other hand, distinctness proved beneficial, but regularizing for sparsity too did not lead to improved performance.

The ideas developed in this thesis could be extended in a number of ways.

Applications of the spectral norm constraint. This thesis proposed a *spectral norm* constraint as a convex penalty to promoting distinctness between a set of vectors. It is easy to imagine other settings where this kind of penalty may be desirable – for example, an fMRI multi-task regression or multi-class classification problem in which one has reason to believe that the ROIs predictive for different tasks (regression) or classes (classification) are spatially disjoint in the brain. Researchers in the machine learning community have experimented with *determinantal point processes* [27] as probabilistic prior distributions that encourage diversity between a set of objects. A spectral norm constraint or penalty is an alternative which fits nicely into the framework of convex optimization.

Theory of the spectral norm constraint. The degree to which the spectral norm constraint enforces distinctness in the following regression problem:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}|| \quad \text{subject to} \quad ||\mathbf{M}||_2 \leq c \tag{44}$$

is controlled by c, the bound on the spectral norm. Distinctness is only enforced when c is sufficiently low. Our empirical studies suggest that the solution to (44) is exactly

$$c\,\mathbf{V}\mathbf{U}^T \quad \text{where} \quad \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{Y}^T\mathbf{W}$$

whenever c is equal to the smallest singular value of the least-squares solution $(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{Y}$. (This also implies that the solution is exactly orthogonal.) However, we were unable to prove this fact.

The wedge penalty. This thesis also explored a "wedge" penalty that promotes distinctness between a set of vectors by penalizing the ℓ_1 norms of their inner products. It would be interesting to develop a version of the wedge penalty that also promotes sparsity, perhaps via an ℓ_1 penalty. It would also be valuable to have better algorithms for optimizing the wedge penalty — subgradient descent is slow and highly sensitive to the step size hyperparameter. This optimization problem is of course complicated by the fact that the wedge is not convex in its most useful parameterizations.

Sparsity and exact orthogonality. To simultaneously encourage distinctness and sparsity, we relaxed a non-convex optimization problem over the set of scaled orthonormal matrices into a convex optimization problem over the set of matrices with bounded spectral norm, so that we could apply generic convex optimization algorithms. Of course, this relaxation comes at a cost — the resulting spatial maps are not *guaranteed* to be anywhere near orthogonal (though in practice they usually are). An alternative approach would be to promote sparsity, perhaps via an ℓ_1 penalty, while maintaining the exact (though non-convex) orthogonality constraint:

$$\arg\min_{\mathbf{M}} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}|| + \alpha||\mathbf{M}||_1 \quad \text{subject to} \quad \mathbf{M}\mathbf{M}^T = \mathbf{I} \tag{45}$$

The set $\{\mathbf{M} : \mathbf{M}\mathbf{M}^T = \mathbf{I}\}$ is known as the *Stiefel manifold*, and optimization over matrix manifolds in general [1] and over the Stiefel manifold in particular [41, 28] have been well-studied. However, far less work has been done on optimizing *non-smooth* functions such as (45) over matrix manifolds, with the notable exception of [26]. A useful avenue of future research would be to explore algorithms that optimize non-smooth functions over the Stiefel manifold, perhaps by constructing a local smooth approximation, as is done in the proximal gradient method.

TFA SRM. The SRM penalties studied in this thesis that performed best on the matching experiment turned out to be those that yielded spatial maps supported on a small and spatially contiguous brain region. A different constraint that would also produce this kind of spatial map would be to require that each spatial map take the form of a radial basis function parameterized by a center location $\boldsymbol{\mu} \in \mathbb{R}^3$ and a radius $\lambda \in \mathbb{R}^+$:

$$m_v = \exp\left(-\frac{||\mathbf{x}_v - \boldsymbol{\mu}||}{\lambda}\right)$$

where $m_v \in \mathbb{R}$ is the value of the spatial map $\mathbf{m}$ at voxel v, and $\mathbf{x}_v \in \mathbb{R}^3$ is the location of voxel v in 3D physical space.

In the fully unsupervised matrix factorization case (i.e. without the assumption of a synchronized stimulus that forces a shared response between subjects), this method has been proposed in the neuroscience literature as *topographic factor analysis* [30]. The ensuing optimization problem is non-convex, but similar optimization problems have been studied for decades as radial basis function networks [5].

A Alternate Algorithms for Learning Timecourses

As we saw above, learning the shared timecourses in a linear shared response model involves solving the optimization problem:

$$\arg\min_{\mathbf{W}} \quad ||\mathbf{Y} - \mathbf{M}\mathbf{W}||_F^2 \quad \text{subject to} \quad ||\mathbf{w}_k|| \leq 1 \tag{46}$$

Our final implementation used a commercial optimization package, Mosek, to solve this problem. However, this approach requires $\mathbf{W}$ to be at most medium scale, and for the user to have certain optimization software installed on their system. In this section we describe two alternative approaches which do not suffer from these limitations: block coordinate descent, and solving the Lagrange dual problem.

A.0.4 Block Coordinate Descent

The approach proposed in [**?**] is to carry out block coordinate descent on the columns of $\mathbf{W}$, solving for one timecourse at a time while holding the others fixed. Letting $\mathbf{w}_k$ denote the k-th column of $\mathbf{W}$, the update is:

$$\mathbf{w}_k \leftarrow \mathbf{w}_k + \frac{2(\mathbf{Y} - \mathbf{W}\mathbf{M})\,\mathbf{m}_k}{||\mathbf{m}_k||_2^2}$$

where $\mathbf{m}_k$ is the k-th row of $\mathbf{M}$ (the k-th spatial map).

A.0.5 Lagrange Dual

Another solution, proposed in [29], is to solve the Lagrange dual of (46).

Consider the so-called *Lagrangian* function of (46):

$$\mathcal{L}(\mathbf{W}, \boldsymbol{\lambda}) = ||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \sum_{k=1}^{K} \lambda_k \left(||\mathbf{w}_k|| - 1\right) \tag{47}$$

The Lagrange dual function is:

$$\mathcal{D}(\boldsymbol{\lambda}) = \min_{\mathbf{W}} \mathcal{L}(\mathbf{W}, \boldsymbol{\lambda}) \tag{48}$$

and the Lagrange dual problem is:

$$\arg\max_{\boldsymbol{\lambda}} \quad D(\boldsymbol{\lambda}) \quad \text{subject to} \;\; \lambda_k \geq 0 \tag{49}$$

The strong duality theorem guarantees that the optimal value of the Lagrange dual problem is equal to the optimal value of our original problem.

Of course, to optimize the Lagrange dual, we need a closed form for the RHS of (48). Given fixed $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_K)$, then simple matrix calculus shows that the minimizer of the Lagrangian with respect to $\mathbf{W}$ has the following closed form solution:

$$\hat{\mathbf{W}} = \arg\min_{\mathbf{W}} \mathcal{L}(\mathbf{W}, \boldsymbol{\lambda}) = \mathbf{Y}\mathbf{M}^T(\mathbf{M}\mathbf{M}^T + \boldsymbol{\Lambda})^{-1} \tag{50}$$

Therefore, the closed form of the Lagrange dual is:

$$\begin{aligned} \mathcal{D}(\boldsymbol{\lambda}) &= \min_{\mathbf{W}} \mathcal{L}(\mathbf{W}, \boldsymbol{\lambda}) \\ &= \mathcal{L}(\hat{\mathbf{W}}, \boldsymbol{\lambda}) \\ &= \text{Tr}(\mathbf{Y}^T\mathbf{Y}) - \text{Tr}(\mathbf{Y}\mathbf{M}^T(\mathbf{M}\mathbf{M}^T + \boldsymbol{\Lambda})^{-1}\mathbf{M}\mathbf{Y}^T) - \text{Tr}(\boldsymbol{\Lambda}) \end{aligned} \tag{51}$$

Therefore, we need to solve:

$$\arg\max_{\boldsymbol{\lambda}} \quad -\text{Tr}(\mathbf{Y}\mathbf{M}^T(\mathbf{M}\mathbf{M}^T + \boldsymbol{\Lambda})^{-1}\mathbf{M}\mathbf{Y}^T) - \text{Tr}(\boldsymbol{\Lambda}) \quad \text{subject to} \;\; \lambda_k \geq 0 \tag{52}$$

Notice that this dual only has K variables, whereas the primal had NK.

The gradient of (52) is given in [29] as:

$$\frac{\partial \mathcal{D}(\boldsymbol{\lambda})}{\partial \lambda_k} = ||\mathbf{Y}\mathbf{M}^T(\mathbf{M}\mathbf{M}^T + \boldsymbol{\lambda})^{-1}e_k||_2^2 - 1$$

After solving (52), plugging $\hat{\boldsymbol{\lambda}}$ into (50) gives the minimizer of the primal problem.

We used the LBFGS algorithm implemented in Scipy as `fmin_l_bfgs_b` to solve (52).

B ADMM for Spectral + ℓ_1

In this appendix we derive an Alternating Direction Method of Multipliers algorithm to solve (**??**):

$$\underset{\mathbf{M}}{\arg\min} \quad \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 + \alpha\,||\mathbf{M}||_1 \quad \text{subject to} \quad ||\mathbf{M}||_2 \leq c$$

ADMM solves problems of the form:

$$\begin{aligned} \underset{\mathbf{X},\mathbf{Z}}{\arg\min} \quad & f(\mathbf{X}) + g(\mathbf{Z}) \\ \text{subject to} \quad & \mathbf{A}\mathbf{X} = \mathbf{B}\mathbf{Z} \end{aligned}$$

where f and g are convex functions.

ADMM introduces an auxiliary variable $\mathbf{U}$ and iterates between these three updates:

$$\mathbf{X}^{t+1} \leftarrow \underset{\mathbf{X}}{\arg\min} \left(f(\mathbf{X}) + \frac{\rho}{2}||\mathbf{A}\mathbf{X} - \mathbf{B}\mathbf{Z}^t + \mathbf{U}^t||_F^2 \right) \tag{53}$$

$$\mathbf{Z}^{t+1} \leftarrow \underset{\mathbf{Z}}{\arg\min} \left(g(\mathbf{Z}) + \frac{\rho}{2}||\mathbf{A}\mathbf{X}^{t+1} - \mathbf{B}\mathbf{Z} + \mathbf{U}^t||_F^2 \right) \tag{54}$$

$$\mathbf{U}^{t+1} \leftarrow \mathbf{U}^t + \mathbf{A}\mathbf{X}^{t+1} - \mathbf{B}\mathbf{Z}^{t+1} \tag{55}$$

We may apply ADMM to our problem by setting:

$$\mathbf{Z} = \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} \mathbf{I}_K \\ \mathbf{I}_K \end{bmatrix} \qquad \mathbf{B} = \mathbf{I}_{2K}$$

$$\begin{aligned} f(\mathbf{X}) &= \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{X}||_F^2 \\ g\left(\begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix}\right) &= g_1(\mathbf{Z}_1) + g_2(\mathbf{Z}_2) \\ g_1(\mathbf{Z}_1) &= \alpha||\mathbf{Z}_1||_1 \\ g_2(\mathbf{Z}_2) &= \mathbf{1}_{||\mathbf{Z}_2||_2 \leq c} \end{aligned}$$

This amounts to imposing the sparsity penalty g_1 on $\mathbf{Z}_1$ alone and the distinctness penalty g_2 on $\mathbf{Z}_2$ alone, but requiring $\mathbf{Z}_1$, $\mathbf{Z}_2$, and $\mathbf{X}$ to all equal each other:

$$\begin{aligned} \underset{\mathbf{X},\mathbf{Z}}{\arg\min} \quad & \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{X}||_F^2 + \alpha||\mathbf{Z}_1||_1 + \mathbf{1}_{||\mathbf{Z}_2||_2 \leq c} \\ \text{subject to} \quad & \mathbf{X} = \mathbf{Z}_1 = \mathbf{Z}_2 \end{aligned}$$

With these choices, equation (53) becomes:

$$\begin{aligned}
\mathbf{X}^{t+1} &\leftarrow \arg\min_{\mathbf{X}} \left(f(\mathbf{X}) + \frac{\rho}{2} ||\mathbf{A}\mathbf{X} - \mathbf{B}\mathbf{Z}^t + \mathbf{U}^t||_F^2 \right) \\
&= \arg\min_{\mathbf{X}} \left(f(\mathbf{X}) + \frac{\rho}{2} \left\| \begin{bmatrix} \mathbf{X} \\ \mathbf{X} \end{bmatrix} - \begin{bmatrix} \mathbf{Z}_1^t \\ \mathbf{Z}_2^t \end{bmatrix} + \begin{bmatrix} \mathbf{U}_1^t \\ \mathbf{U}_2^t \end{bmatrix} \right\|_F^2 \right) \\
&= \arg\min_{\mathbf{X}} \left(f(\mathbf{X}) + \frac{\rho}{2} ||2\mathbf{X} - \mathbf{Z}_1^t - \mathbf{Z}_2^t + \mathbf{U}_1^t + \mathbf{U}_2^t||_F^2 \right) \\
&= \arg\min_{\mathbf{X}} \left(f(\mathbf{X}) + 2\rho \left\| \mathbf{X} - \frac{1}{2}(\mathbf{Z}_1^t + \mathbf{Z}_2^t - \mathbf{U}_1^t - \mathbf{U}_2^t) \right\|_F^2 \right) \\
&= \mathbf{prox}_{\frac{1}{2\rho} f} \left(\frac{1}{2}(\mathbf{Z}_1^t + \mathbf{Z}_2^t - \mathbf{U}_1^t - \mathbf{U}_2^t) \right)
\end{aligned}$$

Equation (54) becomes:

$$\begin{aligned}
\begin{bmatrix} \mathbf{Z}_1^{t+1} \\ \mathbf{Z}_2^{t+1} \end{bmatrix} &\leftarrow \arg\min_{\mathbf{Z}_1, \mathbf{Z}_2} \left(g(\mathbf{Z}) + \frac{\rho}{2} ||\mathbf{A}\mathbf{X}^{t+1} - \mathbf{B}\mathbf{Z} + \mathbf{U}^t||_F^2 \right) \\
&= \arg\min_{\mathbf{Z}_1, \mathbf{Z}_2} \left(\begin{bmatrix} g_1(\mathbf{Z}_1) \\ g_2(\mathbf{Z}_2) \end{bmatrix} + \frac{\rho}{2} \left\| \begin{bmatrix} \mathbf{X}^{t+1} \\ \mathbf{X}^{t+1} \end{bmatrix} - \begin{bmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{U}_1^t \\ \mathbf{U}_2^t \end{bmatrix} \right\|_F^2 \right) \\
&= \begin{bmatrix} \arg\min_{\mathbf{Z}_1} \; g_1(\mathbf{Z}_1) + \frac{\rho}{2} ||\mathbf{X}^{t+1} - \mathbf{Z}_1 + \mathbf{U}_1^t||_F^2 \\ \arg\min_{\mathbf{Z}_1} \; g_2(\mathbf{Z}_2) + \frac{\rho}{2} ||\mathbf{X}^{t+1} - \mathbf{Z}_2 + \mathbf{U}_2^t||_F^2 \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{prox}_{\frac{1}{\rho} g_1} \left(\mathbf{X}^{t+1} + \mathbf{U}_1^t \right) \\ \mathbf{prox}_{\frac{1}{\rho} g_2} \left(\mathbf{X}^{t+1} + \mathbf{U}_2^t \right) \end{bmatrix}
\end{aligned}$$

And finally, equation (55) becomes:

$$\begin{bmatrix} \mathbf{U}_1^{t+1} \\ \mathbf{U}_2^{t+1} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{U}_1^t + \mathbf{X}^{t+1} - \mathbf{Z}_1^{t+1} \\ \mathbf{U}_2^t + \mathbf{X}^{t+1} - \mathbf{Z}_2^{t+1} \end{bmatrix}$$

C Proofs

Theorem C.1. *The following optimization problem has an analytical solution:*

$$\begin{aligned}
&\arg\min_{\mathbf{M}} \; \frac{1}{2} ||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 \quad \text{subject to} \quad \mathbf{M}\mathbf{M}^T = c^2 \, \mathbf{I} \\
&= c \, \mathbf{V}\mathbf{U}^T \quad \text{where} \quad \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{Y}^T\mathbf{W}
\end{aligned}$$

Proof. First, we prove a lower bound on the optimal value. Then, we prove that the solution $\hat{\mathbf{M}} = c \, \mathbf{V}\mathbf{U}^T$ attains this lower bound.

Notice that

$$\begin{aligned}\arg\min_{\mathbf{M}} ||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2 &= \arg\min_{\mathbf{M}} \ \text{Tr}(\mathbf{Y}^T\mathbf{Y}) - 2\text{Tr}(\mathbf{Y}^T\mathbf{W}\mathbf{M}) + \text{Tr}(\mathbf{W}^T\mathbf{W}\mathbf{M}\mathbf{M}^T) \\ &= \arg\min_{\mathbf{M}} \ \text{Tr}(\mathbf{Y}^T\mathbf{Y}) - 2\text{Tr}(\mathbf{Y}^T\mathbf{W}\mathbf{M}) + c^2\,\text{Tr}(\mathbf{W}^T\mathbf{W}) \\ &= \arg\min_{\mathbf{M}} \ -2\text{Tr}(\mathbf{Y}^T\mathbf{W}\mathbf{M}) \\ &= \arg\max_{\mathbf{M}} \ \text{Tr}(\mathbf{Y}^T\mathbf{W}\mathbf{M})\end{aligned}$$

By Von Neumann's trace inequality [25],

$$\begin{aligned}\text{Tr}(\mathbf{Y}^T\mathbf{W}\mathbf{M}) &\leq \boldsymbol{\sigma}(\mathbf{Y}^T\mathbf{W}) \cdot \boldsymbol{\sigma}(\mathbf{M}) \\ &= \text{diag}(\boldsymbol{\Sigma}) \cdot \boldsymbol{\sigma}(\mathbf{M})\end{aligned}$$

Any matrix that satisfies the constraint $\mathbf{M}\mathbf{M}^T = c^2\,\mathbf{I}$ will have all of its singular values at exactly c. Therefore we may conclude that

$$\text{Tr}(\mathbf{Y}^T\mathbf{W}\mathbf{M}) \leq c\,\text{Tr}(\boldsymbol{\Sigma})$$

The solution $\hat{\mathbf{M}} = c\,\mathbf{V}\mathbf{U}^T$ matches this bound:

$$\begin{aligned}\text{Tr}(\mathbf{Y}^T\mathbf{W}\hat{\mathbf{M}}) &= c\,\text{Tr}(\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{U}^T) \\ &= c\,\text{Tr}(\boldsymbol{\Sigma})\end{aligned}$$

Therefore, $\hat{\mathbf{M}}$ is an optimal solution.

□

Theorem C.2. *The proximal operator of* $f(\mathbf{M}) = \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{M}||_F^2$ *is:*

$$\textbf{prox}_{\lambda f}(\mathbf{M}) = \left(\mathbf{W}^T\mathbf{W} + \frac{1}{\lambda}\mathbf{I}\right)^{-1}\left(\mathbf{W}^T\mathbf{Y} + \frac{1}{\lambda}\mathbf{M}\right) \tag{56}$$

Proof.

$$\textbf{prox}_{\lambda f}(\mathbf{M}) = \arg\min_{\mathbf{X}} \ \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{X}||_F^2 + \frac{1}{2\lambda}||\mathbf{M} - \mathbf{X}||_F^2$$

Dropping the $\frac{1}{2}$ term, the RHS expands out to:

$$\text{Tr}(\mathbf{Y}^T\mathbf{Y}) - 2\text{Tr}(\mathbf{Y}^T\mathbf{W}\mathbf{X}) + \text{Tr}(\mathbf{X}^T\mathbf{W}^T\mathbf{W}\mathbf{X}) + \frac{1}{\lambda}\left[\text{Tr}(\mathbf{M}^T\mathbf{M}) - 2\text{Tr}(\mathbf{M}^T\mathbf{X}) + \text{Tr}(\mathbf{X}^T\mathbf{X})\right]$$

Dropping terms that do not depend on $\mathbf{M}$ and grouping together like terms:

$$-2\,\text{Tr}\left(\left(\mathbf{W}^T\mathbf{Y} + \frac{1}{\lambda}\mathbf{M}\right)^T\mathbf{X}\right) + \text{Tr}\left(\left(\mathbf{W}^T\mathbf{W} + \frac{1}{\lambda}\mathbf{I}\right)\mathbf{X}\mathbf{X}^T\right)$$

Since this is a convex function, we may find a global minimizer by setting the gradient to zero and solving for $\mathbf{M}$. The gradient with respect to $\mathbf{M}$ is:

$$\nabla_{\mathbf{M}} = -2\left(\mathbf{W}^T\mathbf{Y} + \frac{1}{\lambda}\mathbf{M}\right) + 2\left(\mathbf{W}^T\mathbf{W} + \frac{1}{\lambda}\mathbf{I}\right)\mathbf{X}$$

The gradient is equal to zero at:

$$\hat{\mathbf{M}} = \left(\mathbf{W}^T\mathbf{W} + \frac{1}{\lambda}\mathbf{I}\right)^{-1}\left(\mathbf{W}^T\mathbf{Y} + \frac{1}{\lambda}\mathbf{M}\right)$$

□

Theorem C.3. *The convex hull of the set of matrices whose rows have norm c and are orthogonal to each other is the set of matrices with spectral norm no larger than c:*

$$\text{conv}(\{\mathbf{M} \in \mathbb{R}^{m\times n} : \mathbf{M}\mathbf{M}^T = c^2\mathbf{I}\}) = \{\mathbf{M} \in \mathbb{R}^{m\times n} : ||\mathbf{M}||_2 \leq c\} \tag{57}$$

Proof. Let $\mathcal{O} = \{\mathbf{M} : \mathbf{M}\mathbf{M}^T = c^2\mathbf{I}\}$ and let $\mathcal{B} = \{\mathbf{M} : ||\mathbf{M}||_2 \leq c\}$. We need to show two facts:

1. any convex combination of elements in $\mathcal{O}$ lies in $\mathcal{B}$.
2. any element in $\mathcal{B}$ is a convex combination of elements in $\mathcal{O}$.

For the first direction, let $\mathbf{M} = \sum_i \alpha_i \mathbf{O}_i$ be a convex combination of matrices $\mathbf{O}_i \in \mathcal{O}$. We claim that each $||\mathbf{O}_i|| = c$. Why? Well, let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the SVD of $\mathbf{O}_i$. Then $\mathbf{O}_i\mathbf{O}_i^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}\mathbf{U}^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T$. Since $\mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T = c^2\mathbf{I}$, one valid choice of $\mathbf{\Sigma}^2$ is with c^2 as each element on the diagonal. Therefore, all of the singular values of $\mathbf{O}_i$ are exactly c, and so is its spectral norm. Therefore, by the triangle inequality,

$$\left\|\sum_i \alpha_i \mathbf{O}_i\right\|_2 \leq \sum_i \alpha_i \|\mathbf{O}_i\|_2 = \sum_i \alpha_i c = c$$

For the second direction, let $\mathbf{M}$ be any matrix such that $||\mathbf{M}||_2 \leq c$. Let $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the SVD of $\mathbf{M}$, and let $\boldsymbol{\sigma} = \text{diag}(\mathbf{\Sigma})$. Since $||\mathbf{M}||_2 \leq c$, we have $\boldsymbol{\sigma} \in [0, c]^n$.

Consider the 2^n points $\{0, c\}^n$, which are the corners of the rectangle $[0, c]^n$. Since $[0, c]^n$ is the convex hull of $\{0, c\}^n$, $\boldsymbol{\sigma}$ is a convex combination of these points. By lemma (C.4) below, $\boldsymbol{\sigma}$ is also a convex combination of the points $\mathbf{x}_1 \ldots \mathbf{x}_{2^n} = \{-c, c\}^n$:

$$\boldsymbol{\sigma} = \sum_{i=1}^{2^n} \alpha_i \mathbf{x}_i$$

Now define $\mathbf{O}_i = \mathbf{U}\,\text{diag}(\mathbf{x}_i)\mathbf{V}^T$. Notice that

$$\mathbf{O}_i\mathbf{O}_i^T = \mathbf{U}\mathbf{\Sigma}^2\mathbf{U}^T = c^2\mathbf{U}\mathbf{U}^T = c^2\mathbf{I}$$

Hence $\mathbf{O}_i \in \mathcal{O}$.

We complete the proof by showing that $\mathbf{M}$ is a convex combination of the matrices $\{\mathbf{O}_i\}_{i=1}^{2^n}$.

$$\begin{aligned}
\sum_{i=1}^{2^n} \alpha_i \mathbf{O}_i &= \sum_{i=1}^{2^n} \alpha_i \mathbf{U}\,\text{diag}(\mathbf{x}_i)\mathbf{V}^T \\
&= \mathbf{U}\,\text{diag}\left(\sum_{i=1}^{2^n} \alpha_i\ \mathbf{x}_i\right)\mathbf{V}^T \\
&= \mathbf{U}\,\text{diag}\left(\boldsymbol{\sigma}\right)\mathbf{V}^T \\
&= \mathbf{M}
\end{aligned}$$

Lemma C.4. *If* $\mathbf{a}$ *is a convex combination of* $\{0, c\}^n$*, then* $\mathbf{a}$ *is also a convex combination of* $\{-c, c\}^n$*.*

Let $\mathbf{v}_1 \dots \mathbf{v}_{2^n} = \{0, c\}^n$ and suppose that

$$\mathbf{a} = \sum_{i=1}^{2^n} \alpha_i \, \mathbf{v}_i \tag{58}$$

Define $\mathbf{c} = \{c\}^n$ and define $\{\mathbf{u}_i\}_{i=1}^{2^n}$ as

$$u_{ij} = \begin{cases} +c & \text{if } v_{ij} = c \\ -c & \text{if } v_{ij} = 0 \end{cases}$$

Notice that $\mathbf{c}, \mathbf{u}_i \in \{-c, c\}^n$.

We claim that

$$\mathbf{a} = \frac{1}{2}\mathbf{c} + \frac{1}{2} \sum_{i=1}^{2^n} \alpha_i \, \mathbf{u}_i \tag{59}$$

Why? Well, by (58), we have

$$a_j = c \sum_{i:\, v_{ij}=c} \alpha_i$$

And by (C), we have

$$\begin{aligned} a_j &= \frac{1}{2}\, c\, 2^n + \frac{1}{2} \left[c \sum_{i:\, v_{ij}=c} \alpha_i - c \sum_{i:\, v_{ij}=0} \alpha_i \right] \\ &= \frac{1}{2} \left[c \sum_{i:\, v_{ij}=c} \alpha_i + c \sum_{i:\, v_{ij}=0} \alpha_i \right] + \frac{1}{2} \left[c \sum_{i:\, v_{ij}=c} \alpha_i - c \sum_{i:\, v_{ij}=0} \alpha_i \right] \\ &= c \sum_{i:\, v_{ij}=c} \alpha_i \end{aligned}$$

Finally, we notice that

$$\frac{1}{2} + \frac{1}{2} \sum_i \alpha_i = \frac{1}{2} + \frac{1}{2} = 1$$

Therefore, $\mathbf{a}$ is a convex combination of the points $\{-c, c\}^n$. □

References

[1] PA Absil, Robert Mahoney, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds.* 2008.

[2] Christian Beckmann and Stephen Smith. "Probabilistic Independent Component Analysis for Functional Magnetic Resonance Imaging." *IEEE Transactions on Medical Imaging*. 2004.

[3] Christian Beckmann. "Modelling with Independent Components." *Neuroimage.* 2012.

[4] Dmitri Bertsekas. *Nonlinear Programming.* 1999.

[5] Christopher Bishop. *Neural Networks for Pattern Recognition.* 1995.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* 2004.

[7] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." *Foundations and Trends in Machine Learning.* 2010.

[8] Guiseppe Calafiore and Laurent El Ghaoui. *Optimization Models.* 2014.

[9] Vince Calhoun, Vamsi Potluru, Ronald Phlypo, Rogers Silva, Barak Pearlmutter, Arvind Caprihan, Sergey Plis, and Tulay Adali. "Independent Component Analysis for Brain fMRI Does Indeed Select for Maximal Independence." *PLOS ONE.* 2013.

[10] J. Chen, C.J. Honey, E. Simony, M.J. Arcaro, K.A. Norman, and U. Hasson. "Accessing Real-Life Episodic Information from Minutes versus Hours Earlier Modulates Hippocampal and High-Order Cortical Dynamics." *Cerebral Cortex.* 2015.

[11] J. Chen, Y.C. Leong, K.A. Norman, and U. Hasson. "Shared Experience, Shared Memory: A Common Structure for Brain Activity During Naturalistic Recall." *Preprint.* 2015.

[12] Po-Hsuan Chen, Janice Chen, Yaara Yeshurun-Dishon, Uri Hasson, James Haxby, Peter Ramadge. "A Reduced-Dimension fMRI Shared Response Model." *Neural Information Processing Systems*. 2015.

[13] I. Daubechies, E. Roussos, S. Takerkart, M. Benharrosh, C. Golden, K. D'Ardenne, W. Richter, J.D. Cohen, and J. Haxby. "Independent Component Analysis for Brain fMRI Does Not Select for Independence." *Proceedings of the Natural Academy of the Sciences.* 2009.

[14] John Duchi, Elad Hazan, and Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization." *Journal of Machine Learning Research.* 2011.

[15] James Haxby, M. Ida Gobbini, Maura Furey, Alumit Ishai, Jennifer Schouten, and Pietro Pietrini. "Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex" *Science*. 2001.

[16] James Haxby, J. Swaroop Guntapalli, Andrew Connoly, Yaroslav Halchenko, Bryan Conroy, M. Ida Gobbini, Michael Hanke, and Peter Ramadge. "A Common, High-Dimensional Model of the Representational Space in Human Ventral Temporal Cortex." *Neuron*. 2011.

[17] Uri Hasson, Yuval Nir, Ifat Levy, Galit Fuhrmann, and Rafael Malach. "Intersubject Synchronization of Cortical Activity During Natural Vision." *Science*. 2004.

[18] Alexander Huth, Thomas Griffiths, Frederick Theunissen, and Jack Gallant. "PrAGMATiC: a Probabilistic and Generative Model of Areas Tiling the Cortex." ArXiv. 2015.

[19] Logan Grosenick, Brad Klingenberg, Kiefer Kantovich, Brian Knutson, and Jonathan E. Taylor. "Whole-brain Prediction Analysis with GraphNet." *Neuroimage*. 2013.

[20] Gene Golub and Charles van Loan. "Matrix Computations." 1996.

[21] Amin Jalali, Lin Xiao, and Maryam Fazel. "Variational Gram Functions: Convex Analysis and Optimization." *arXiv*. 2015.

[22] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning* 2010.

[23] James Haxby, J. Swaroop Guntupali, Andrew Conolly, Yaroslav Halchenko, Bryan Conroy, M. Ida Gobbini, Michael Hanke, and Peter Ramadge. *A Common, High-Dimensional Model of the Representational Space in Human Ventral Temporal Cortex*. 2011.

[24] Mohamed Hebiri and Sara A. Van De Geer. "The Smooth-Lasso and other $\ell_1 + \ell_2$-penalized methods." *Electronic Journal of Statistics*. 2011.

[25] Roger Horn and Charles Johnson. *Matrix Analysis*. 1990.

[26] Artiom Kovnatsky, Klaus Glashoff, and Michael M. Bronstein. "MADMM: A Generic Algorithm for Non-Smooth Optimization on Manifolds." *ArXiv*. 2015.

[27] Alex Kulesza and Ben Taskar. "Determinantal Point Processes for Machine Learning." *Foundations and Trends in Machine Learning.* 2012.

[28] Rongjie Lai and Stanley Oscher. "A Splitting Method for Orthogonality Constrained Problems." *Journal of Scientific Computing*. 2014.

[29] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng. "Efficient sparse coding algorithms." *Neural Information Processing Systems*. 2007.

[30] Jeremy Manning, Rajesh Ranganath, Kenneth Norman, and David Blei. "Topographic Factor Analysis: A Bayesian Model for Inferring Brain Networks from Neural Data." *PLOS ONE.*

[31] Kenneth Norman, Sean Polyn, Gret Detre, and James Haxby. "Beyond mind-reading: multi-voxel pattern analysis of fMRI data." *Trends in Cognitive Sciences.* 2006.

[32] Neal Parikh and Stephen Boyd. "Proximal Algorithms." *Foundations and Trends in Optimization*. 2013.

[33] Hugo Raguet, Jalal Fadili, and Gabriel Peyre. "A Generalized Forward-Backward Splitting." *SIAM Journal of Imaging Sciences.* 2013.

[34] Mert Sabuncu, Benjamin Singer, Bryan Conroy, Ronald Bryan, Peter Ramadge, and James Haxby. "Function-based Intersubject Alignment of Human Cortical Anatomy." *Cerebral Cortex*. 2010.

[35] Naum Z. Shor. *Minimization Methods for Non-differentiable Functions.* 1985

[36] Robert Tibshirani. "Regression shrinkage and selection via the lasso." *Journal of the Royal Statistical Society, Series B.* 1994.

[37] Gael Varoquaux, Alexandre Gramfort, Fabian Pedregosa, Vincent Michel, and Bertrand Thirion. "Multi-subject dictionary learning to segment an atlas of brain spontaneous activity." *Information Processing in Medical Imaging*. 2011.

[38] Gael Varoquaux and Bertrand Thirion. "Statistical Learning for Resting-State fMRI: Successes and Challenges." *NIPS Workshop on Maching Learning and Interpretation in Neuroimaging.* 2011.

[39] Kevin Vervier, Pierre Mahe, Alexandre D'Aspremont, Jean-Baptiste Veyieras, and Jean-Philippe Vert. "On Learning Matrices with Orthogonal Columns or Disjoint Supports." *Machine Learning and Knowledge Discovery in Databases.* 2014.

[40] David Weiss. "Probabilistic Additive Component Analysis." *Princeton senior thesis.* 2007.

[41] Zaiwen Wen and Wotao Yin. "A Feasible Method for Optimization with Orthogonality Constraints." *Mathematical Programming.* 2013.

[42] Dengyong Zhou, Lin Xiao, and Mingrui Wu. "Hierarchical Classification via Orthogonal Transfer." *International Conference on Machine Learning.* 2011.

[43] Hui Zou and Trevor Hastie. "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society, Series B.* 2005.