Factor Topographic Latent Source Analysis: Factor Analysis for Brain Images*

Jeremy R. Manning[1,2], Samuel J. Gershman[3], Kenneth A. Norman[1,3], and David M. Blei[2]

[1] Princeton Neuroscience Institute, Princeton University
[2] Department of Computer Science, Princeton University
[3] Department of Psychology, Princeton University
manning3@princeton.edu

Abstract. Traditional approaches to analyzing experimental functional magnetic resonance imaging (fMRI) data entail fitting per-voxel parameters to explain how the observed images reflect the thoughts and stimuli a participant experienced during the experiment. These methods implicitly assume that voxel responses are independent and that the unit of analysis should be the voxel. However, both of these assumptions are known to be untrue: it is well known that voxel activations exhibit strong spatial correlations, and common sense tells us that the true underlying brain activations are independent of the resolution at which the brain image happened to be taken. Here we propose a fundamentally different approach, whereby brain images are represented as weighted sums of spatial functions. Our technique yields compact representations of the brain images that leverage spatial correlations in the data and are independent of the image resolution.

1 Introduction

Functional magnetic resonance imaging (fMRI) has revolutionized the field of cognitive neuroscience: this technique allows researchers to take high resolution 3-dimensional snapshots of an experimental participant's brain activity approximately once per second throughout an experiment. We present a novel factor analysis-like method for representing and examining brain images obtained from fMRI experiments. Our technique yields low-dimensional representations of the brain images that can be manipulated, compared, and examined more easily than the original images.

In early approaches to analyzing fMRI data, researchers examined univariate differences in the average brain images recorded during different experimental conditions [3, 4, 11]. The fundamental assumption driving this "contrast-based" approach was that voxel-by-voxel differences between images recorded during different conditions should reflect brain structures whose absolute activations vary between the conditions. Modern approaches eschew contrast-based experimental designs in favor of multivariate pattern analysis (MVPA) techniques for detecting patterns that distinguish between experimental variables of interest (e.g. the category of the word that a participant is thinking about as each image is taken). Unlike univariate approaches, multivariate approaches are sensitive to the relative activations between regions (for review see [9]).

Multivariate methods entail fitting per-voxel parameters to explain how the observed images reflect the different conditions in the experiment. This general approach has two primary shortcomings. First, because the per-voxel parameters are fit independently, traditional approaches do not leverage the strong spatial correlations known to pervade fMRI images. Second, fitting per-voxel parameters means that the unit of analysis is determined by the resolution of the brain images, rather than the underlying neuroanatomy.

Topographic Latent Source Analysis (TLSA; [6]) offers a fundamentally different approach to analyzing brain images. TLSA models brain images as linear combinations of spatial functions, called *sources*. Fitting parameters for sources (rather than voxels) removes the assumption that voxels are independent. For example, one can account for spatial autocorrelations in the brain images by using spatial functions that spread their

* We thank Sean Gerrish for useful discussions. This work was supported by National Science Foundation Grant 1009542.

mass over nearby locations. In addition, if the spatial functions are continuous, the representations that TLSA yields will be independent of the resolution(s) of the original brain images.

The TLSA model assumes that the source weights vary as a function of the experimental covariates associated with each image. Here the term 'covariate' refers to the thoughts and stimuli the participant is experiencing as each image is collected. This supervised approach is useful for inferring the neural substrates that support the representations of those experimental covariates. Here we present a related *unsupervised* approach, called Factor-TLSA (F-TLSA), that decomposes brain images into latent sources that are not constrained by the experimental covariates. In this way, F-TLSA is a type of factor analysis for brain images.

In our implementation, we assume that sources are specified by radial basis functions. If a radial basis function has center μ and width λ, its activation $f(\mathbf{r}|\mu, \lambda)$ at location $\mathbf{r}$ is given by

$$f(\mathbf{r}|\mu, \lambda) = \exp\left\{\frac{||\mathbf{r} - \mu||^2}{\lambda}\right\}. \quad (1)$$

Given the observed brain images, our goal is to infer the center location μ_k and width λ_k of each of K sources. We would also like to infer the degree to which each source is active (i.e., weighted) in each of N observed images. Once we have determined sources' locations and widths, the K-dimensional vector of fitted weights provides a reduced representation of the brain image (assuming that $K < V$, where V is the number of voxels in the image).

Using fMRI images to infer the parameterizations of the latent sources assumed by F-TLSA poses substantial computational challenges. This is because fMRI datasets can be very large: each fMRI image typically contains many thousands of voxels, and often hundreds or thousands of images are collected over the course of an experiment. We use a scalable stochastic variational inference algorithm to fit these latent sources. This allows us to fit the parameters of thousands of latent sources to large fMRI datasets using modest hardware (e.g. a laptop computer). One of the primary challenges to analyzing fMRI data is that the datasets are large and exceedingly complicated. Thus we see the low-dimensional representations afforded by F-TLSA as providing a useful tool to the neuroscientific community.

2 Methods

In this section we begin by developing the notation used throughout the remainder of this paper, and by using the notation to describe F-TLSA formally in the probabilistic graphical modeling sense. We then describe an innovative method that we found useful for initializing the parameter estimates. Finally, given the initial parameter estimates, we describe how we use Stochastic Variational Optimization (SVO; [5]), a method for performing stochastic variational inference [7], to further refine the parameter estimates.

2.1 Notation

Let N be the number of observed brain images, K be the number of latent sources whose parameters we wish to infer, and V be the number of voxels in each D-dimensional brain image (for standard fMRI images, $D = 3$). The model consists of the variables summarized in Table 1 (all are real-valued scalars, unless otherwise specified).

2.2 Generative Process

F-TLSA assumes that observed brain images arise from the following generative process (Fig. 1A):

1. For each of K sources:
   (a) Pick source locations $\mu_k \sim \mathcal{N}(\mathbf{c}, \sigma_s^2 \mathbf{I}^D)$ (where $\mathbf{c}$ is the center of the brain).
   (b) Pick source widths $\lambda_k \sim \mathcal{N}(\alpha_s, \beta_s)$.
2. For each of N trials:
   (a) Pick source weights $w_{nk} \sim \mathcal{N}(\alpha_w, \beta_w)$.
   (b) Pick voxel activations $y_{nv} \sim \mathcal{N}\left(\sum_{k=1}^{K} w_{nk} f_v(\mu_k, \lambda_k), \sigma_n^2\right)$ (where σ_n^2 is a voxel noise parameter).

Table 1. Variables in F-TLSA.

Variable	Description
y_{nv}	Voxel v's activation on the n^{th} trial. We use $\mathbf{y}_n$ to refer to the V-dimensional vector of voxel activations on trial n. When the subscript is omitted (e.g. $\mathbf{y}$), this denotes the full set of images, $\mathbf{y}_{1\ldots N}$.
w_{nk}	The activation of the k^{th} source during trial n. We use $\mathbf{w}_n$ to refer to the K-dimensional vector of source activations on trial n. When the subscripts are omitted (e.g. $\mathbf{w}$), this denotes the full set of source activation (weight) vectors, $\mathbf{w}_{1\ldots N}$.
$\mu_k \in \mathbb{R}^D$	The center of the k^{th} source (μ_{kd} is the coordinate in the d^{th} dimension).
λ_k	The width of the k^{th} source.
$f_v(\mu, \lambda)$	The basis image, specified by center μ and width λ, evaluated at the location of voxel v. We use $\mathbf{F}$ to refer to the matrix of K (unweighted) basis images, specified by $\mu_{1\ldots K}, \lambda_{1\ldots K}$, where the k^{th} row corresponds to the basis images for the k^{th} source.

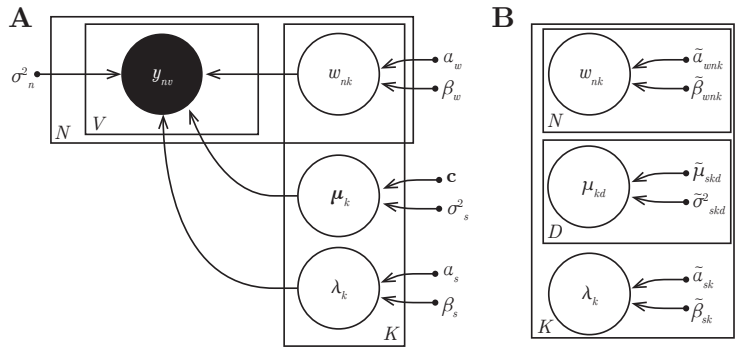

Fig. 1. A. Graphical model. The D-dimensional pattern of V voxel activations is observed during each of N trials (y_{nv}). The model assumes that each of K fixed sources are specified by centers (μ_k) and widths (λ_k). The voxel activations arise due to the sources being activated to varying amounts on each trial, as specified by w_{nk}. (Shaded nodes indicate observed variables, unshaded nodes indicate hidden variables, and dots indicate hyperparameters.) **B. Variational approximation.** We assume that the source weights, centers, and widths are independent and are governed by their own variational parameters (indicated by $\sim$'s).

2.3 Parameter fitting

We use variational inference to fit the hidden variables of F-TLSA given the observed brain images. The general idea is that we will posit a variational distribution q over the hidden variables, and tune the variational parameters of q to minimize the Kullback-Leibler (KL) divergence between q and the posterior distribution p over the hidden variables given the data defined by F-TLSA. This poses the inference problem as an optimization problem. Because variational inference is notoriously sensitive to the starting point of the parameter estimates, we first describe our procedure for initializing the variational parameters, and then we describe how we tune those estimates using variational inference.

2.4 Initialization

Given the observed brain images, our goal is to infer the center location μ_k and width λ_k of each source (which remain fixed across images), and the vector of source weights $\mathbf{w}_n$ for each image.

Initializing source centers and widths. We initialize source centers and widths using an iterative process, illustrated in Figure 2. We place each source, one at a time, at locations in the image with high levels of activation, termed *hotspots*. After placing each source, we grow its width (starting from a small width) until the source explains the associated hotspot in the image (Fig. 2A). We then create a source image

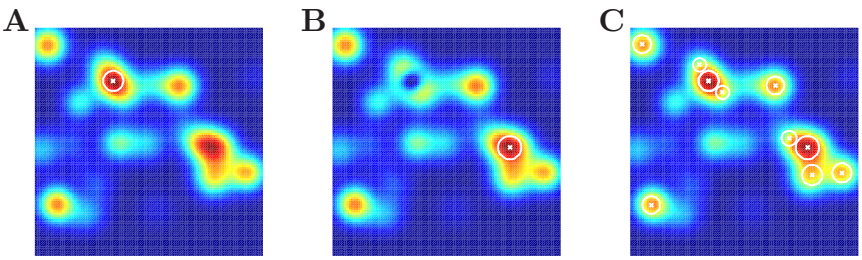

Fig. 2. Initializing source centers and widths. Here we illustrate the initialization procedure for a synthetic 2-dimensional example image. **A. Fitting the first source.** We begin by placing the first source's center at the location at which the image displays maximal activation (indicated by the white $\times$). We then grow the width of the source until it explains the given hotspot in the image (the white level curve, which indicates the locations at which the source's value is 0.5, is used to illustrate the fitted source's width). **B. Fitting subsequent sources.** Subsequent sources are fit using the same procedure, but on the residual image (after previous sources have been subtracted off). Here the source localized and sized using the original image has been subtracted off, leaving a "hole" in the image. The next hotspot appears at a different location, as shown. **C. The full procedure.** The process of iteratively fitting sources to the residual images continues until $K = 10$ sources are placed. Note that the original synthetic image was constructed using 25 sources, and thus some regions of the image are not explained by the fitted sources.

by evaluating the activation of the source (given the center and width parameters) at the location of each voxel in the brain image. We subtract the source image from the brain image; the resulting residual image contains the brain activations that are left unexplained by the source. We then fit the next source's location and width using the residual image (Fig. 2B). This process of fitting sources to the residual brain images continues until K sources (with K specified in advance) are placed (Fig. 2C).

To fit the source widths, we need a means of (a) detecting the extent of a given hotspot in the image, and (b) determining when a source is wide enough to explain the hotspot. To accomplish this, we define an image threshold ρ; all voxels with activations above the ρ^{th} percentile are set to 1 in a thresholded brain image, and all other voxels are set to 0. We also define a source threshold ϕ; all voxels where the source has a value greater than ϕ are set to 1 in a thresholded source image, and all other voxels are set to 0. Starting with a source width of 1 voxel, we incrementally grow the source by 1 voxel until the proportion of voxels set to 1 in the thresholded source image that are also set to 1 in the thresholded brain image falls below 95%. Having identified the source's location and width, we subtract that source from the (non-thresholded) brain image, and repeat the process of initializing source locations and widths for the subsequent sources using the residual brain image. We found that setting $\rho = 90$ and $\phi = 0.5$ works well in practice.

Initializing source weights. F-TLSA models voxel activations as draws from Gaussian distributions whose means are linear combinations of latent sources. In expectation, the vector of voxel activations is given by

$$\mathbf{y}_n = \mathbf{w}_n \mathbf{F}. \tag{2}$$

We know $\mathbf{y}_n$ (i.e., the n^{th} observed brain image). If we also knew $\mathbf{F}$ (i.e., the matrix of unweighted source activations), then we could solve for the expected source weights by multiplying both sides by $\mathbf{F}^{-1}$. We can estimate $\mathbf{F}$ by computing the expected source locations and widths with respect to $q(\mu_{1\ldots K,1\ldots D})$ and $q(\lambda_{1\ldots K})$. Given this estimate of $\mathbf{F}$, we can solve for each $\mathbf{w}_n$ using the associated observed brain image.

2.5 Tuning the parameter estimates

After initializing the variational parameters, we use a variational inference algorithm to further refine the parameter estimates. We want to compute the probability of the hidden variables, $\mathbf{x} = \{w_{1\ldots N,1\ldots K}, \mu_{1\ldots K}, \lambda_{1\ldots K}\}$, given the observed data, $\mathbf{y}$. Once we compute $p(\mathbf{x}|\mathbf{y})$, we can use this posterior distribution to set these hidden variables (e.g. to the most probable values, to the expected values, etc.).

Variational inference. If we have already specified our generative model $p(\mathbf{y}|\mathbf{x})$ and a prior over the hidden variables $p(\mathbf{x})$, and if we knew the prior probability of observing each possible value of the data $p(\mathbf{y})$, then we could use Bayes' rule to compute $p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y})}$. However, computing $p(\mathbf{y}) = \int p(\mathbf{x}, \mathbf{y})d\mathbf{x}$ is usually intractable, because it would require integrating over all possible settings of the hidden variables in the model. However, we can compute $p(\mathbf{x}|\mathbf{y})$ up to a normalization constant by ignoring the $p(\mathbf{y})$ term. The second issue is that if the parameterized forms of $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x})$ are non-conjugate (i.e., their product does not have the same parametric form as $p(\mathbf{x})$), then the resulting form of $p(\mathbf{x}|\mathbf{y})$ will often be prohibitively difficult to work with and/or intractable. Instead of computing $p(\mathbf{x}|\mathbf{y})$ directly, which is difficult or impossible for the two above reasons, variational methods make use of a simpler distribution, $q(\mathbf{x}|\theta)$, that is easy to work with by design. The idea is that one can tune the variational parameters θ, governing the form of $q(\mathbf{x}|\theta)$, until $q(\mathbf{x}|\theta)$ is closest to $p(\mathbf{x}, \mathbf{y})$ as measured by the KL divergence between the two distributions. This is often accomplished by tuning θ to maximize the evidence lower bound (ELBO), $\mathcal{L}_\theta$ (see [1] for further discussion):

$$\mathcal{L}_\theta = \mathbb{E}_{q(\theta)}[\log p(\mathbf{x}, \mathbf{y}) - \log q(\mathbf{x}|\theta)]. \tag{3}$$

Stochastic Optimization of the Variational Objective (SVO). SVO [5] is a form of stochastic variational inference [7, 8, 10]. We first draw M samples, $\mathbf{x}_{1\ldots M}$ from $q(\theta)$. We then use those samples to approximate the gradient of the ELBO with respect to θ, $\nabla\mathcal{L}_\theta$:

$$\nabla_\theta \mathcal{L}_\theta \Big|_{\theta_{t-1}} \approx \frac{1}{M} \sum_{m=1}^{M} \nabla_\theta \log q(\mathbf{x}_{tm}|\theta_{t-1}) \left(\log p(\mathbf{x}_{tm}, \mathbf{y}) - \log q(\mathbf{x}_{tm}|\theta_t)\right), \tag{4}$$

where the subscript t refers to the number of iterations of SVO we have run. Given this approximate gradient, we use coordinate ascent to iteratively update each element of θ. If we can also compute the second gradient, $\frac{\partial}{\partial^2}\mathcal{L}_\theta$, then we can perform more accurate (second-order) updates of θ.

Unlike traditional variational inference algorithms, SVO does not require us to derive update equations for θ. Rather, we simply need to be able to compute $\log p(\mathbf{x}, \mathbf{y})$ and $\log q(\mathbf{x}|\theta)$ up to normalization constants, $\nabla_\theta \log q(\mathbf{x}|\theta)$, and (if we wish to perform second order updates) $\frac{\partial}{\partial\theta^2} \log q(\mathbf{x}|\theta)$. We also need to be able to sample from $q(\mathbf{x}|\theta)$ given an estimate of θ (see [5] for details).

Fitting the hidden variables, $\mathbf{x} = \{w_{1\ldots N,1\ldots K}, \mu_{1\ldots K}, \lambda_{1\ldots K}\}$, of F-TLSA using SVO requires first constructing a variational distribution over the hidden variables in p, $q(w_{1\ldots N,1\ldots K}, \mu_{1\ldots K,1\ldots D}, \lambda_{1\ldots K})$. For simplicity, we constructed $q(\mathbf{x})$ to fully factorize, as shown in Figure 1B. Our goal is to adjust the variational parameters, $\theta = \{\tilde{\alpha}_{w,1\ldots N,1\ldots K}, \tilde{\beta}_{w,1\ldots N,1\ldots K}, \tilde{\mu}_{s,1\ldots K,1\ldots D}, \tilde{\lambda}_{s,1\ldots K}\}$, of $q(\mathbf{x})$ to bring $q(\mathbf{x})$ into best alignment with $p(\mathbf{x}, \mathbf{y})$. Whereas in the full model $p(\mathbf{x}, \mathbf{y})$ the hidden variables are drawn from Gaussian distributions, our variational distribution $q(\mathbf{x})$ is not restricted to Gaussians:

- The source weights for each image, w_{nk}, are drawn from truncated Gaussians (lower bound at 0, no upper bound) with mean $\tilde{\alpha}_{wnk}$ and variance $\tilde{\beta}_{wnk}$.
- The source centers in each dimension, $\tilde{\mu}_{kd}$, are drawn from truncated Gaussians (lower and upper bounds determined by the range of voxel locations with the given dimension, d) with mean $\tilde{\mu}_{skd}$ and variance $\tilde{\sigma}^2_{skd}$.
- The source widths are drawn from truncated Gaussians (lower bound at 0, no upper bound) with mean $\tilde{\alpha}_{sk}$ and variance $\tilde{\beta}_{sk}$.

We would like to obtain the set of source locations, widths, and weights that best explain the observed set of images. The variational parameters that govern these variables are $\tilde{\alpha}_{w,1\ldots N,1\ldots K}$, $\tilde{\mu}_{s,1\ldots K,1\ldots D}$, and $\tilde{\alpha}_{s,1\ldots K}$. We leave the parameters governing the variances of the corresponding truncated Gaussian distributions fixed.

After initializing the source locations and widths using the above procedure, we use SVO to adjust the initial estimates of the source locations and widths using the average brain image (as during initialization). There are two main reasons why using the average brain image to fit the source locations and widths makes sense. First, the source locations and widths are assumed to be constant across brain images (Fig. 1C).

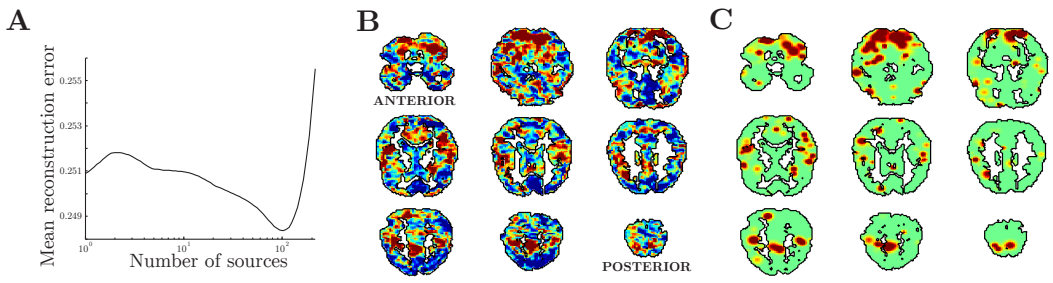

Fig. 3. Reconstructions. A. Reconstruction error as a function of K. Errors are averaged across 3,863 images (comprising 529,892 voxels) from 11 participants, predicting activations of 1,000 randomly held-out voxels per image (repeated 25 times per image, for each value of K). **B. Example image from one participant**. **C. Reconstructed image using $K = 100$ sources.**

Thus the source locations and width should reflect cohesive "blobs" of activity that are present across many brain images. These blobs will be easier to identify in the mean image, because it will be less noisy than the individual images (since elements not in common across many images will be averaged out). Second, fitting source locations and widths to a single (mean) brain image will be much faster than fitting source locations and widths to the hundreds (or thousands) of images collected during a given experiment.

After honing the source locations and widths, we next update the estimates of the source weights for each image. Because the source weights within a given image are assumed to be independent of the source weights in other images, we can save processing time (by a factor of N) by considering each individual image in isolation. We begin by initializing the source weights as described above. We then loop through each source, using SVO to update each source's weight. After performing n_{iter} updates for each source, we move on to the next image.

3 Results

To assess the quality of F-TSLA's representations, we used the above procedures to fit source locations and widths using $N_{training} < N$ brain images collected as participants viewed a series of words on a computer screen. We then fit the source weights for each of the $N_{test} = N - N_{training}$ remaining images. However, rather than using the full images, we sub-sampled $V_{training} < V$ voxels from each of the images. We then computed the mean squared difference between the actual voxel activations of the remaining $V_{test} = V - V_{training}$ voxels in each image to the activations predicted by the reduced F-TLSA representation. This provided a measure of how well the F-TLSA representations generalized across brain images from a given participant, and of how well the representations could account for missing data (sometimes referred to as the *in-painting problem* [2]). We found that reconstruction errors varied with the number of sources, with the minimum error occurring when we used $K = 100$ sources (Fig. 3A). We show sample brain images from a single participant, and the corresponding reconstructions, in Figures 3B,C.

4 Concluding remarks

We have presented an unsupervised method for representing fMRI images as weighted sums of spatial sources. Our technique may be easily implemented (and modified) without requiring the practitioner to derive complicated variational updates. The representations are independent of the resolution of the original brain images, and elegantly capture spatial autocorrelations in the data. In addition, the representations can be several orders of magnitude more compact than the original images (depending on the desired number of sources, K). Our preliminary explorations suggest that the representations nonetheless retain general spatial features of the original images. We envision these reduced representations as providing the neuroscientific community with a useful means of exploring, comparing, and understanding complex neural images.

Bibliography

[1] Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

[2] Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on image processing*, 15(12):3736 – 3745.

[3] Friston, K., Holmes, A., Worsley, K., Poline, J.-P., Frith, C., and Frackowiak, R. (1995). Statistical parameter maps in functional imaging: a general linear approach. *Human Brain Mapping*, 2:189 – 210.

[4] Friston, K., Price, C., Fletcher, P., Moore, C., Frackowiak, R., and Dolan, R. (1996). The trouble with cognitive subtraction. *NeuroImage*, 4:97 – 104.

[5] Gerrish, S. and Blei, D. M. (2012). Stochastic optimization of the variational bound. *Submitted*.

[6] Gershman, S., Norman, K., and Blei, D. (2011). A topographic latent source model for fMRI data. *NeuroImage*, 57:89 – 100.

[7] Hoffman, M., Blei, D., Wang, C., and Paisley, J. (2012). Stochastic variational inference. *ArXiv e-prints*, 1206.7051.

[8] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37:183 – 233.

[9] Norman, K. A., Polyn, S. M., Detre, G. J., and Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fMRI data. *Trends in Cognitive Sciences*, 10(9):424–430.

[10] Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Foundations and trends in machine learning*, 1(1-2):1 – 305.

[11] Zarahn, E., Aguirre, G., and D'Esposito, M. (1997). A trial-based experimental design for fMRI. *NeuroImage*, 6:122 – 138.