

# **Probabilistic Additive Component Analysis**

A Latent Variable Model for Dimensionality Reduction of  
Human Functional Magnetic Resonance Images

David Weiss

A thesis submitted in partial satisfaction  
of the requirements for the degree of

Bachelor of Arts

Department of Computer Science,  
Princeton University

Advisors:

Ken Norman, Psychology  
David Blei, Computer Science

May 7, 2007

## Abstract

In recent years, an important new application of machine learning research has emerged from the field of cognitive neuroscience. In ‘mind-reading’ experiments, a machine learning classifier is trained to predict aspects of a human subject’s mental state from patterns of brain activity recorded by in a functional MRI (fMRI) scanner. However, a typical fMRI dataset consists of relatively few, noisy observations of brain patterns consisting of tens of thousands of individual spatial features (“voxels”), so feature selection or dimensionality reduction is required in order for classification to succeed.

In this thesis, we present a novel method of dimensionality reduction that incorporates three key neuroscientific assumptions into a probabilistic, generative model of fMRI data, which we call Probabilistic Additive Component Analysis (PACA). We provide an algorithm to fit the model using maximum a posteriori (MAP) estimation, and we show analytically that MAP estimation is equivalent to matrix factorization with L2 and L1 regularizations and a non-negativity constraint on one of the factors. We then compare PACA against two similar dimensionality reduction methods—principal component analysis (PCA) and non-negative matrix factorization (NMF)—analytically and by running each algorithm experimental fMRI data from two cognitive neuroscience experiments.

We find that both PCA and NMF satisfy one but violate two of the neuroscientific assumptions motivating PACA, and that PACA outperforms both PCA and NMF using both unsupervised and supervised measures of the quality of the reduced data. However, supervised methods of feature selection resulted in higher overall classification accuracy than unsupervised methods in all but an synthetic, “idealized” case. We conclude that PACA demonstrates that the integration of existing knowledge into a generative model can improve analysis of fMRI data, and several new practical and theoretical improvements on the model are proposed.

## Acknowledgments

In accordance with the MIT License Agreement, credit is hereby given to the Massachusetts Institute of Technology and to the Center for Biological and Computational Learning for providing the database of facial images used in this thesis (hereby referred to in this thesis as the “CBCL Face Library.”)

I would like to thank everyone without whom this thesis would have been impossible. Had not my advisor Ken Norman took me under his wing in the Fall of 2004, I would not be writing this thesis today. I am also deeply grateful to my advisor Dave Blei for his careful attention, helpful suggestions, patience, and guidance in the preparation of this thesis. I am thankful to Denis Chirigev and Ehren Newman for taking time after lab meeting for long and helpful discussions, and to Susan Robison and Matt Weber for allowing me access to their research data (even though some of that data didn’t make it into the thesis). I’m also completely indebted to Greg Detre for innumerable helpful instructions and spur-of-the-moment technical meetings, and for always taking time to help me get moving again when I got stuck. I would like to thank my parents for their unconditional love and support through the most difficult parts of the last year and a half, and my brothers for helping me proofread at a moment’s notice.

Finally, I would like to thank most of all my future wife and best friend Gilli, without whom none of this would have been possible.

### **Pledge of Honor**

I hereby declare that this thesis represents my own work in accordance with Princeton University regulations.

Signed,

David Weiss

# Contents

Abstract . . . . .	ii
Acknowledgments . . . . .	iii
Pledge of Honor . . . . .	iv
<b>1 Introduction</b>	<b>1</b>
1.1 Decoding Mental State from Brain Activity . . . . .	2
1.2 Latent Variable Models for Dimensionality Reduction . . . . .	4
1.3 Probabilistic Graphical Models . . . . .	7
1.4 <i>Maximum A Posteriori (MAP)</i> Estimation . . . . .	12
1.5 Related Work . . . . .	14
<b>2 Probabilistic Additive Component Analysis (PACA)</b>	<b>16</b>
2.1 Motivation . . . . .	17
2.2 Model Definition . . . . .	21
2.2.1 Notation . . . . .	21
2.2.2 Generative Process . . . . .	22
2.2.3 Probabilistic Graphical Model and Full Joint Distribution . . . . .	25
2.3 Estimating $\mu$ and $\mathbf{Z}$ from Data . . . . .	28
2.3.1 Unconstrained Optimization . . . . .	30
2.4 Setting Hyperparameters . . . . .	31
2.4.1 Minimization of Squared Reconstruction Error . . . . .	32
2.4.2 L2 Regularization of Topic Means . . . . .	33
2.4.3 L1 Regularization of Brain States . . . . .	34
2.4.4 Alternative Formulation: Regularized Matrix Factorization . . . . .	35
2.4.5 Setting hyperparameters for Proportional Regularization . . . . .	36
2.5 Comparison to Existing Methods . . . . .	38
2.5.1 Principal Component Analysis (PCA) . . . . .	38
2.5.2 Non-negative Matrix Factorization (NMF) . . . . .	39
2.6 Implementation . . . . .	41
2.6.1 PACAfit and PACApredict . . . . .	41
2.6.2 PCA and NMF . . . . .	41
2.6.3 Running Experiments . . . . .	42
2.6.4 Initialization & Convergence . . . . .	43

<b>3</b>	<b>Results</b>	<b>44</b>
3.1	CBCL Face Library . . . . .	44
3.1.1	Convergence . . . . .	45
3.1.2	“Additive Face Components” . . . . .	45
3.2	Human fMRI Datasets . . . . .	50
3.2.1	Face and Object Representations (Haxby et al., 2001) . . . . .	52
3.2.2	Retrieval Orientation State Memory (Robison et al., 2007) . . . . .	53
3.2.3	Synthetic Datasets . . . . .	54
3.3	Assessing Generalization Using Cross Validation . . . . .	54
3.3.1	Avoiding Sparse Brain States . . . . .	58
3.3.2	Reconstruction Error . . . . .	58
3.3.3	Classifier Prediction Error . . . . .	61
3.3.4	Determining “Best Case” Improvements with Synthetic Data . . . . .	67
3.4	Visualizing the Reduced Data . . . . .	68
3.4.1	Neural Topics . . . . .	68
3.4.2	Brain States . . . . .	69
<b>4</b>	<b>Discussion</b>	<b>74</b>
4.1	Key Points . . . . .	75
4.1.1	Modeling Assumptions are Important . . . . .	75
4.1.2	Supervised vs. Unsupervised Analysis . . . . .	76
4.1.3	Choosing the Right Number of Components . . . . .	76
4.1.4	Benefits of Regularization . . . . .	77
4.1.5	Limitations of Experimental Results . . . . .	78
4.2	Directions for Future Work . . . . .	79
	<b>References</b>	<b>81</b>

# Chapter 1

## Introduction

In the analysis of any sufficiently large dataset, it is often necessary to reduce an immense, noisy set of complicated measurements into a few succinct conclusions about the underlying processes generating the observed data. For example, one might be interested in drawing conclusions about global weather patterns from local weather station measurements, tracking the location of an aircraft measured only by noisy radar (Russell & Norvig, 2003), or inferring research topics from a corpus of documents based on word frequencies (Blei et al., 2003). In this thesis, we focus on the problem of ‘mind-reading’: the decoding of a human subject’s mental state from fMRI measurements of brain activity over time (Norman et al., 2006; Haynes & Rees, 2006). The ‘mind-reading’ problem has become of increasing interest within the last few years, as the ability to read out even measures of cognitive state has allowed the direct testing of psychological theories of memory that were previously impossible (Polyn et al., 2005). However, the ‘mind-reading’ problem is difficult, because fMRI datasets often contain tens of thousands of uninformative features and the signal-to-noise ratio is very low Mitchell et al. (2004).

This thesis seeks to address the problem of *feature selection*, the necessary preprocessing stage of any ‘mind-reading’ experiment in which uninformative input features are removed from the analysis. We propose a novel latent variable model of fMRI data that incorporates existing neuroscientific knowledge in the form of three critical generative assumptions. We call this model *Probabilistic Additive Component Analysis (PACA)*. Using

this model, estimates of the latent variables replace the original feature set and reduce the dimensionality of the original data in a multivariate fashion; the reduced data is expressed in terms of stereotyped patterns of excitation and inhibition, or “neural topics.” To test the model, we show experimentally that the new model results in representations of brain state that improve generalization performance in ‘mind-reading’ type experiments relative to two contemporary methods of dimensionality reduction, PCA and NMF.

In this chapter, we first provide background information describing the current state of ‘mind-reading’ research and the reasons why the analysis is technically challenging (section 1.1). We then give a brief introduction to latent variable methods for dimensionality reduction, which represent an alternative to traditional methods of feature selection (section 1.2). In section 1.3, we give a brief introduction to the framework of probabilistic graphical models, which we will use in chapter 2 for the specification of the PACA model. In section 1.4, we describe the general methods used to find estimates of unobserved variables in probabilistic models, which we will use in chapter 2 to find estimates of the latent variables of the PACA model. Finally, in section 1.5, we briefly discuss other research in ‘mind-reading’ fMRI analysis that pursues ideas relevant or similar to those presented in this thesis.

## **1.1 Decoding Mental State from Brain Activity**

In a typical ‘mind-reading’ experiment, a human subject performs a cognitive task while in a functional Magnetic Resonance Imaging (fMRI) scanner, and the goal of subsequent analysis is to predict aspects of the cognitive task from the recorded patterns of brain activity (Norman et al., 2006). The tasks involved vary widely across experiments, depending on aspects of cognition that the experimenter wishes to study. Machine learning algorithms have been trained to recognize patterns of brain activity associated with recall of specific visual memories (Polyn et al., 2005), the category of images viewed by a subject (Cox &

Savoy, 2003), whether a subject was lying to the experimenter during the task (Davatzikos et al., 2005), and many others (Norman et al., 2006). There is similar diversity in the variety of analysis techniques applied to the data to generate a prediction rule. Most commonly, machine learning classification algorithms such as a neural networks for logistic regression (Norman et al., 2006), support vector machines (Mitchell et al., 2004), or Fischer Linear Discriminants (Haynes & Rees, 2005) are used in the analysis, but presumably any sufficiently powerful classification algorithm will suffice (Norman et al., 2006).

However, there remain significant challenges to any mind-reading fMRI experiment that have not yet been overcome, and first and foremost among these is the problem of feature selection. A typical fMRI dataset contains  $500 \sim 1000$  recorded timepoints of brain activity, but each timepoint is comprised of  $10^4 \sim 10^5$  voxels, or “volume pixels”, depending on the spatial resolution of the scanner. Using all  $10^5$  voxels as inputs to a generic classification algorithm is often infeasible. Furthermore, the activity of only a small fraction of the potentially  $10^5$  voxels provides information that is relevant to the task at hand, due to neuroanatomical constraints and the high level of noise in fMRI recordings. Thus, even if it were practical to use the entire brain as the input to the classifier, it would be undesirable to do so. Instead, it is necessary to select only those input features that are likely to improve analysis of a given experiment. We can narrow down the number of voxels by restricting our analysis to brain regions we expect to be involved in the cognitive task, but feature selection is still a difficult problem.

Due to the difficult nature of the problem, there is no “standard” approach to feature selection, but the Statistical Parametric Mapping (SPM) approach is relatively common (Norman et al., 2006). SPM uses a statistical test to assign a value for each voxel indicating its association with the cognitive variables of interest. The basic procedure for SPM involves two steps: calculating a uni-variate test statistic for each voxel, such as an ANOVA p-value or F-score, and then selecting the top  $V$  voxels according to the test statistic for use in the rest of the analysis (Shen & Meyer, 2006). The choice of  $V$  can be based on either

prior expectations (like the anatomical masks) or through an empirical method like cross validation (section 3.3.2).

However, there are two important limitations to the SPM method. First, SPM attempts to find the collection of voxels that *collectively* predict cognitive state by grouping together the voxels that are most informative *individually*; only choosing voxels that individually correlate with or predict the cognitive variables of interest might incorrectly exclude voxels that are encoding brain state as part of a population, and ultimately the goal of the analysis is to classify *patterns* of brain activity. Ideally, feature selection would involve choosing pattern-based features that do not rely on uni-variate assessments of individual voxels (Norman et al., 2006). The second major limitation of SPM for feature selection is that it is a *supervised* method. Supervised methods require that the data and the labels that are present. Thus, SPM cannot be used on unlabeled data points, even though such points will carry information about the dataset. Ideally, a partially supervised or completely unsupervised method for feature selection would use all of the available datapoints to produce the inputs for the classifier. In the next section, we discuss potential means for finding just such a method.

## 1.2 Latent Variable Models for Dimensionality Reduction

Rather than seeking to answer the question, “Which of the input features are the most useful for our analysis?”, we can alternatively ask, “Which *transformation* of these features is the most useful for our analysis?” Ideally, we could map from the high-dimensional voxel space to a much smaller set of features that would contain all the important high-level information in the original voxel set while discarding noise and irrelevant activity. The problem of finding this useful, reduced dimension representation of a given dataset is known as *dimensionality reduction*, and it provides an attractive alternative to traditional feature selection methods. The goal of dimensionality reduction algorithms is to find a fea-

ture set of many lower dimensions that is nonetheless “equivalent” to the original dataset, in the sense that the useful properties of the old feature set can be reconstructed from the new feature set with a certain degree of accuracy. Unlike SPM, dimensionality reduction algorithms are typically both inherently multivariate and unsupervised, satisfying both of the major criticisms of SPM raised above; algorithms for dimensionality will attempt to find a reduced version of the original data without any other knowledge than the datapoints themselves.

In general, there are many (sometimes equivalent) families of dimensionality reduction algorithms, but we focus here on a particularly intuitive and powerful group of methods known as *latent variable models*. Latent variable models are used in many fields and defined in many different but equivalent ways, but they can be summarized as follows: in a latent variable model, one assumes that the observed data was generated according to a specific process governed by a set of unobserved (or *latent*) variables. If one knew the values of the latent variables, the actual observed datapoints would be redundant—all the important aspects of the original data could be recreated using the generative process that we assume has generated the data in the first place. Latent variable models can be used for dimensionality reduction when it is possible to infer the latent variables from the dataset and when the dimensionality of the latent variables is lower than the dimensionality of the original data; we can simply replace the observed data by the inferred values of the latent variables.

An example of latent variable models for dimensionality reduction is Latent Dirichlet Allocation (LDA) (Blei et al., 2003). LDA is a probabilistic model of text documents which can reduce a corpus of documents (represented by vectors of word counts) to probability distributions over “topics,” which are themselves probability distributions over words (Blei et al., 2003). Given a corpus of documents as an input, LDA will find the set of  $K$  topics that best explains the data according to the model. Thus, if there are  $D$  documents with a dictionary of size  $W$  words, the  $D \times W$  input matrix  $\mathbf{X}$  is reduced to a  $D \times K$  matrix

expressing the documents in terms of  $K$  topics, and a  $K \times W$  matrix defining the topics over words. By choosing  $K \ll W$ , LDA will produce matrices of latent variables that are much smaller than the original data, but which can be used to recreate the word counts of the original documents according to the LDA model. In this way, LDA can reduce the dimensionality of an entire corpus of documents to a much more manageable size.

Of course, it is important to remember that just because a model-based dimensionality reduction algorithm will infer the model parameters from a given dataset is no indication that the output will have any meaningful interpretation. In order to successfully reduce the dimensionality of data, any latent variable model must make simplifying assumptions about the origin of the data. The utility of the reduced data produced by the model depends on the degree to which the underlying assumptions are appropriate to both the real origins of the data and the desired interpretations of the reduced data. On the one hand, if the model assumptions are misguided, oversimplified, or is simply incorrect, the resulting reduced data produced by the model will very likely have captured nothing of interest about the original dataset. On the other hand, even if the model does capture important aspects of the data, and even if the model reproduces the data successfully, the simplified representation of the data within the model might be useless for the desired analysis. For instance, although LDA will produce excellent models of documents in terms of the counts of words within the document, LDA would be useless for researchers interested in investigating the grammatical structure of a corpus of documents.

With such a large number of voxels and such a small number of observations, it is likely that there are many ways in which to represent fMRI datasets in terms of smaller sets of latent variables, and it is equally likely that some of these representations are more useful for the ‘mind-reading’ context than others. If we are to use latent variable models for ‘mind-reading’ experiments, it is therefore crucial that the assumptions underlying the model are appropriate for the neuroimaging context. We now consider one representation of statistical models that provides an excellent framework for specifying complex models

from a concise set of straightforward assumptions: probabilistic graphical models.

### 1.3 Probabilistic Graphical Models

How does one construct a model? Often, there are many equivalent ways of expressing a single statistical model, and which representation is the most “natural” is a matter of debate. For our purposes, and throughout this thesis, we will primarily consider *probabilistic graphical models*. Probabilistic graphical models are useful because they provide an intuitive, compact representation of otherwise complex models. Furthermore, given a probabilistic graphical model and a corresponding set of conditional probability distributions, it is easy to find other important equivalent probabilistic representations of the model: namely, the *generative process* and *joint probability distribution*.

In this thesis, we focus on a specific sub-class of probabilistic graphical models, *directed* graphical models. A directed graphical model is a directed acyclic graph (DAG)  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  where each node  $v$  is associated with a random variable  $X_v$ , and where each node  $X_v$  is defined to be conditionally independent of all other nodes given its parents. Recall that conditional Independence, read “ $X$  is conditionally independent of  $Y$  given  $Z$ ,” is defined as

$$P(X, Y|Z) = P(X|Z)P(Y|Z), \quad (1.1)$$

where  $X$ ,  $Y$ , and  $Z$  are random variables. The significance of the conditional independence relationships in the graphical model is that they allow the joint probability distribution of *all* nodes in the model can be written succinctly entirely in terms of conditional probability distributions:

$$P(X) = \prod_{v \in \mathcal{V}} P(X_v | \text{Parents}(X_v)). \quad (1.2)$$

Equation 1.2 is practically useful in the statistical modeling we describe in this thesis for two reasons. First, it is often much easier to specify conditional probability distributions

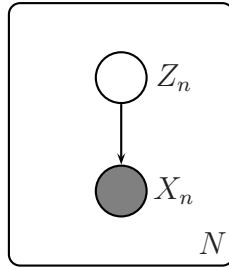


Figure 1.1: A simple graphical model of coin flipping. The unobserved variable  $Z_n$  (open circle) determines the identity of the coin chosen, and thus the probability distribution of the observed flips  $X_n$  (filled circle). We represent this relationship in the graph by a directed edge from  $Z_n$  to  $X_n$ . Rather than draw this diagram for each  $Z_n$  and  $X_n$ , the enclosing plate around the two variables indicates  $N$  repetitions of the structure.

than joint probability distributions when we are describing complex systems. For instance, if we are modeling  $N$  binary variables, such as the occurrences of words in a document, we need to assign probabilities to all  $2^N$  possible combinations of outcomes to describe the full joint distribution, and we might feel unsure as to what these probabilities should be. On the other hand, it's a lot easier to define the conditional probabilities of words appearing in a document given that we know the document is about a certain topic (Blei et al., 2003). The other major benefit of defining graphical models using conditional probability is that many times, many different variables in the model will share the same conditional probability distribution, so equation 1.2 can be simplified even further.

As an example, consider a simple model of a coin-flipping system. The system consists of two steps. First, a coin is chosen at random by the system from a pool of available coins. Some of these coins are fair coins that will land on “heads” or “tails” with equal probability, but some coins are unfair, and these will always land on “heads.” It is equally likely that a fair or unfair coin will be chosen by the system. The second step is to flip the chosen coin  $N$  times. What is the probability of observing 10 “heads” flips in a row? We could attempt to model the system by figuring out the joint probability of every possible outcome: fair coin chosen, flip 1 is “heads”, flip two is “tails”, and so forth. However, the most straightforward answer exploits the local conditional probability relationships defined

by the system through *marginalization*: if the coin was unfair, the probability is 1. If the coin was fair, the probability is  $1/2^{10}$ . Since the probability of choosing each coin is  $1/2$ , the probability of observing 10 “heads” flips in a row is  $1/2 + 1/2^{11}$ .

We can now form a compact, intuitive probabilistic graphical model of the coin flip system. To formalize our knowledge, we define an unobserved random variable  $Z$  to represent the choice of coin, and the observed random variables  $X_1, \dots, X_N$  for each coin flip. Then we define the following local probability distributions:

$$pP(Z = \text{Fair}) = 1/2, \quad P(Z = \text{Unfair}) = 1/2,$$

$$P(X_n = H|Z = \text{Fair}) = 1/2, \quad P(X_n = H|Z = \text{Unfair}) = 1.$$

These state that there is equal probability that the chosen coin fair or unfair, but that an unfair coin will always land on heads. Finally, we can draw an associated graph  $\mathcal{G}$ , with nodes for  $Z$ , and  $X_1$  through  $X_N$  (figure 1.1). To indicate that  $Z$  is an unobserved variable (we don’t know which coin was chosen by the system), the node for  $Z$  is left unfilled. From equation 1.2, we now have the full joint probability distribution for our model:

$$P(X_{1:N}, Z) = P(Z) \prod_{n=1}^N P(X_n|Z). \tag{1.3}$$

Now let us consider how this example relates to latent variable models. Suppose we had observed a sequence of  $N$  coin flips, and we wished to infer which coin had been chosen by the system. By applying Bayes Rule, we know that

$$P(Z|X_{1:N}) = \frac{P(X_{1:N}|Z)P(Z)}{\sum_Z P(X_{1:N}, Z)}. \tag{1.4}$$

Using the conditional probability distributions defined above, we can evaluate the numerator of equation 1.4, and we can evaluate the denominator using equation 1.3. Thus, we can use Bayes rule to infer the value of the latent variable  $Z$  in this model from an observed

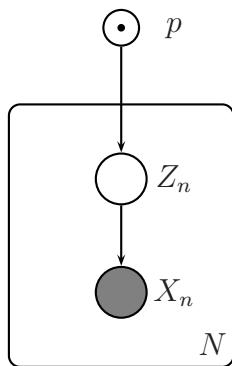


Figure 1.2: A slightly more complicated model of coin flipping. We add a parameter  $p$  that establishes a prior distribution on the coin choices  $Z_n$ . This fixed parameter is notated with a dot in the circle, and because each  $Z_n$  is conditionally independent given  $p$ , the  $p$  node is located outside of the  $N$  plate.

dataset  $X_{1:N}$ . We could even consider replacing  $X_{1:N}$  with the value  $Z$  as a form of dimensionality reduction, since, knowing  $Z$ , we can recreate data with the same distribution as  $X_{1:N}$ .

At this point it is worth considering what the model *cannot* do, due to the limitations of the simplifying assumptions we have made. One important consequence of the way we have defined our model is that the individual coin flips are *exchangeable*: because each coin flip depends only on  $Z$ , the actual ordering of the flips is irrelevant to the model. We see this in the product term of equation 1.4: the ordering of the terms in a product does not change the result, so we can reorder the coin flips however we like. Equivalently, each coin flip is independent of every other coin flip given the unknown quantity  $Z$ . However, this assumption might be inappropriate for the coin flipping system. If we supposed instead that the coins were landing in a malleable surface, such as jello, then this assumption might not hold – the later flips might be biased by the deformities introduced into the jello by the previous flips. Furthermore, from the dimensionality reduction standpoint, exchangeability might limit the usefulness of the model. If we need to recreate  $X_{1:N}$  exactly, preserving order, we will need a more complex model than the one in figure 1.1.

To finish off our simple example of probabilistic graphical models, we increase the

complexity of our model by adding the assumption that the probability of choosing a fair coin varies as the coin flipping experiment is repeated on different days. We introduce a new parameter  $p$ , and change our conditional distribution for  $Z$  as follows:

$$P(Z = \text{Fair}|p) = p, \quad P(Z = \text{Unfair}|p) = 1 - p.$$

Because  $Z$  now depends on  $p$ , we must add another node to our graphical model (figure 1.2). However, since we are not assuming  $p$  to be a random variable, however, we mark the node for  $p$  with a dot; this indicates that  $p$  is a fixed parameter. Suppose we want to infer a value for  $p$  from a set of observed coin flips  $X_{1:N}$ ? Unlike in equation 1.4, there is no probability distribution  $P(p|X_{1:N}, Z)$  that we can calculate. In this case, we must form a point estimate  $\hat{p}$  using a method such as maximum likelihood estimation (section 1.4).

Although we have only looked at a simplistic example, the example illustrates the three primary benefits of the graphical approach that we advocate in this thesis. First, all the assumptions of the model are immediately accessible and easy to understand, as conditional probability distributions must be specified for each random variable in the model. This makes it easy to qualitatively assess the suitability of the model to a given problem (e.g., the “jello problem”), and to qualitatively compare one model to another. Second, the conditional probability distributions are sufficient to fully specify the entire model, so probabilistic graphical models are easy to build, modify, and then expand later (e.g., adding the parameter  $p$ ). Finally, once the probabilistic model is fully specified, we can apply any number of standard algorithms to form estimates of unknown variables or parameters, such as maximum likelihood estimation or expectation maximization (EM).

Finally, the last section of background material we cover is the specific method of approximate inference used in the remainder of this thesis: *maximum a posteriori* (MAP) estimation.

## 1.4 *Maximum A Posteriori (MAP) Estimation*

Let us now consider any probabilistic model in a somewhat more abstract case. Given a set of observed data  $\mathcal{D}$  and a set of governing parameters  $\theta$ , the probability of observing the data under a particular parameter set is defined as the *likelihood*  $l$  of the data with parameters  $\theta$ . For a fixed dataset  $\mathcal{D}$ , the likelihood function is a function of the parameters  $\theta$ :

$$l(\theta; \mathcal{D}) = p(\mathcal{D}|\theta). \quad (1.5)$$

Note that we can define the likelihood function even if  $\theta$  is a set of fixed parameters that have no distribution  $p(\theta)$ . We can use the likelihood function to infer estimates of the parameters from the observed data. Consider the set of parameters  $\hat{\theta}_{\text{ML}}$  that maximize the likelihood function; we can think of these parameters as the parameters that have the “best chance” at explaining the observed data. This is because  $\mathcal{D}$  has the highest likelihood of being observed when  $\theta = \hat{\theta}_{\text{ML}}$ . If we wish to find a point estimate of the parameters, then  $\hat{\theta}_{\text{ML}}$  will be a reasonable choice. We define  $\hat{\theta}_{\text{ML}}$  formally as

$$\hat{\theta}_{\text{ML}} = \underset{\theta}{\operatorname{argmax}} l(\theta; \mathcal{D}), \quad (1.6)$$

and  $\hat{\theta}_{\text{ML}}$  are thus known as *maximum likelihood estimators* (Russell & Norvig, 2003). In practice, it is often easier to maximize the *log likelihood*, which we denote  $L$ :

$$\begin{aligned} \hat{\theta}_{\text{ML}} &= \underset{\theta}{\operatorname{argmax}} \log l(\theta; \mathcal{D}) \\ &= \underset{\theta}{\operatorname{argmax}} L(\theta; \mathcal{D}) \end{aligned} \quad (1.7)$$

If we considered the unobserved variable  $Z$  in the latent variable model in figure 1.2 to be the parameters of the model, we could have chosen to ignore the probability distribution  $P(Z)$  and estimated  $Z$  using  $\hat{Z}_{\text{ML}}$  instead of calculating the distribution in equation 1.4.

However, a more complicated model might also assume a *prior distribution*  $p(\theta)$  over

the parameters. Intuitively, this is assuming that parameter sets  $\theta$  some areas of the parameter space are more likely to be observed than others. If we want to parameterize the prior distribution with its own set of parameters, we write

$$p(\theta) = p(\theta|\eta), \quad (1.8)$$

where  $\eta$  is the set of *hyperparameters* of the model. Under these assumptions, it is not sufficient to estimate  $\theta$  by simply maximizing the likelihood function. For example, consider the case where  $p(\hat{\theta}_{\text{ML}}|\eta)$  is extremely low. In this case, the prior probability distribution is saying that observing the estimator  $\hat{\theta}_{\text{ML}}$  is very unlikely, even though  $\hat{\theta}_{\text{ML}}$  might best explain the observed data. To avoid this problem, we can maximize the *posterior distribution* of the parameters given the observed dataset, rather than the likelihood. The posterior distribution  $p(\theta|\mathcal{D})$  can be defined in terms of the likelihood and the prior distribution using Bayes Theorem:

$$\begin{aligned} p(\theta|\mathcal{D}) &= \frac{p(\mathcal{D}|\theta)p(\theta|\eta)}{p(\mathcal{D})} \\ &\propto p(\mathcal{D}|\theta)p(\theta|\eta) = l(\theta; \mathcal{D})p(\theta|\eta). \end{aligned} \quad (1.9)$$

Instead of maximizing the likelihood, we can instead maximize the posterior  $p(\theta|\mathcal{D})$ ; this is known as *maximum a posteriori (MAP)* estimation (Russell & Norvig, 2003). We define the MAP estimates  $\hat{\theta}_{\text{MAP}}$  as the set of parameters that maximize the posterior:

$$\hat{\theta}_{\text{MAP}} = \underset{\theta}{\operatorname{argmax}} l(\theta; \mathcal{D})p(\theta|\eta). \quad (1.10)$$

In practice, it is often easier to to maximize the log of the posterior distribution, which we define as a function of  $\theta$ .

$$\mathcal{L}(\theta; \mathcal{D}, \eta) = \log l(\theta; \mathcal{D}) + \log p(\theta|\eta), \quad (1.11)$$

In this case, we see that to find  $\hat{\theta}_{\text{MAP}}$ , we maximize the sum of the log likelihood and the log of the prior over the parameters. We also note that this can be written as the sum of the log likelihood of the data and the log likelihood of the parameters:

$$\mathcal{L}(\theta; \mathcal{D}, \eta) = L(\theta; \mathcal{D}) + L(\eta; \theta). \quad (1.12)$$

Thus, MAP estimation is equivalent maximum likelihood estimation with a penalty term  $\log p(\theta|\eta)$  added to emphasize certain portions of the parameter space. Such penalty terms are used elsewhere in areas of statistics as *regularization* (Hastie et al., 2003). We will explore this relationship between MAP estimation and regularization in more detail within the context of the PACA model in section 2.4.4.

## 1.5 Related Work

Although the dimensionality reduction method presented in this thesis is novel, the “multi-voxel” approach to fMRI analysis is a more general method that has been attracting increasing interest over the past few years (Haxby et al., 2001; Polyn et al., 2005; Norman et al., 2006; Detre et al., 2006). Recently, a multi-voxel approach to feature selection has been proposed that uses a “searchlight” to perform statistical tests on spatially localized groups of voxels (Kriegeskorte et al., 2006). However, this method is fundamentally different from the approach advocated in this paper, as the individual features produced by the method are still voxels, and it is fundamentally a fully supervised method.

In general, there have been many prior applications of dimensionality reduction techniques to fMRI datasets, but very few within the context of decoding mental state from brain activity. Most involve the application of Principal Component Analysis (PCA), a powerful technique which has been applied in almost innumerable contexts (we discuss PCA in more detail in section 2.5.1). Fan et al. (2006) used PCA for dimensionality reduction as one part of a three-tiered feature selection scheme. Although not classifying

brain activity, Ford et al. (2003) used Principal Component Analysis (PCA) to reduce the dimensionality of voxel activation maps to classify activation maps as belonging to patients with various neurological diseases. PCA has also been applied to many other fMRI analysis in other experimental paradigms (Pettersson1 et al., 1999; Andersen et al., 1999). Non-negative matrix factorization (NMF), a much newer technique, has been applied to fMRI only recently (Wang et al., 2004). Although we do not discuss Independent Components Analysis (ICA) in this paper, ICA can be used in conjunction with PCA as a method for isolating spatially independent components with fMRI datasets. ICA has been applied to a wide variety of fMRI analyses (McKeown et al., 1998; Formisano et al., 2004). Finally, a wide variety of non-linear dimensionality reduction techniques have been used in paradigms outside of the 'mental state decoding' described here (Shen & Meyer, 2006).

Although the mostly Bayesian and probabilistic approach presented in this thesis has been applied to the analysis of fMRI data before, the use has been largely limited to Bayesian versions of conventional fMRI analysis (Pettersson1 et al., 1999; Frank et al., 1998). Notably, Mitchell and colleagues have developed and applied a directed graphical model, the Hidden Process Model (HPM), to certain limited fMRI datasets within a 'mind-reading' type of context (Mitchell & Rustandi, 2006). However, most of the linear methods for dimensionality reduction described above have been shown to belong to a single family of generative statistical models, but with slightly varying assumptions (Tipping & Bishop, 1999; Roweis & Ghahramani, 1999); thus, the general method described in this thesis has already been widely used, even if it is not yet widely recognized as such. When we compare the PACA model to PCA and NMF in more detail in sections 2.5.1 and 2.5.2, we shall see that PACA is also a member of the same general family of models, but with assumptions that are more appropriate to fMRI datasets.

## Chapter 2

# Probabilistic Additive Component Analysis (PACA)

In this section, we introduce a probabilistic generative model of functional MRI data, which we call *Probabilistic Additive Component Analysis (PACA)*. We first introduce in detail the key motivating assumptions behind the model and compare these assumptions to those behind two linear methods of dimensionality reduction, Principal Component Analysis (PCA) and Non-negative Matrix Factorization (section 2.1). The model is defined in detail in section 2.2, and the procedure for fitting the model from data is described in section 2.3. An alternative formulation of the model as a regularized matrix factorization is given in section 2.4.4, and in sections 2.4.2 and 2.4.3 it is shown that regularization induced by the priors specified in our assumptions are equivalent to the L2 and L1 regularization used in many other algorithms. A principled method for selecting hyper-parameters based on the proportion of regularization desired is presented in section 2.4.5. Finally, a working implementation of PACA is described in section 2.6.

## 2.1 Motivation

Given that there already exists a plethora of linear and non-linear methods for dimensionality reduction, why introduce another linear model? First, as discussed in section 1.2, latent variable models for dimensionality reduction offer an attractive alternative to supervised feature selection methods, but ascertaining and assessing the assumptions underlying these models is critically important when interpreting the results. Therefore, rather than simply applying existing models to the ‘mind-reading’ experiments, and then assessing the validity of the assumptions involved, we set out to create a new model with valid, neuroscientific assumptions from the outset. Furthermore, as discussed in section 1.3, it is relatively easy to construct statistical models using the framework of probabilistic graphical models; this also will allow us to expand the complexity of the model to incorporate spatial or time-series relationships in the future (section 4.2).

The PACA model was designed to capture three specific assumptions (two neuroscientific, and one statistical) that we believe are appropriate for analysis of fMRI datasets. These assumptions are defined as follows:

1. *Brain activity is driven by a finite number of excitatory and inhibitory neural processes, or “neural topics.”* This is a basic assumption that is fairly common across all areas of neuroscience, and evidence for both task-related excitation and task-related inhibition in functional imaging experiments is readily available (McKiernan et al., 2003). We capture this assumption in our model by assuming that each underlying neural topic  $k$  has a stereotyped effect on each voxel  $v$ , which we represent by a real-valued scalar parameter  $\mu_{k,v}$ . Excitation is represented by positive values and inhibition represented by negative values, and a single process can have excitatory and inhibitory effects in different brain areas simultaneously. For example, if there were a neural topic that corresponded to a particular working memory task, we might expect to find high positive values of  $\mu_{k,v}$  in areas of prefrontal cortex associated with

high level cognitive tasks and short-term memory, and negative values of  $\mu_{k,v}$  in areas responsible for the processing of distracting stimuli during the experiment.

2. *Neural topics sum additively to produce the observed voxel activations at any given point in time.* Specifically, the measured activity  $x$  of each voxel  $v$  at each time  $t$  is defined by

$$x_{t,v} = \sum_k z_{k,t} \mu_{k,v} + \epsilon, \quad (2.1)$$

where  $\mu_{k,v}$  is the effect of the  $k$ 'th neural topic on the  $v$ 'th voxel (described above),  $z_{k,t}$  is a positive real number indicating the degree of activation of the  $k$ 'th neural topic at time  $t$ , and  $\epsilon$  is an error term. The linear sum makes our model linear (as we shall see, similar to PCA and NMF) and greatly simplifies the mathematical analysis.

The requirement that the topic activations  $z_{k,t}$  are positive serves two purposes. First, we wish to account for the fact that the cognitive variables being decoded in most ‘mind-reading’ experiments are almost exclusively positive. For example, in a category-classification experiment, where the goal is to predict what class of stimuli a subject is viewing (Cox & Savoy, 2003), the cognitive variable of interest is a binary indicator for whether or not a particular stimulus class is present. Thus, if the stimulus is “dog,” we wish to model brain activity involved in representing the concepts of “dog” vs. “no dog” through the various activations of the unobserved neural topics we are trying to model. If we were to allow negative activation of topics, then by this reasoning we would be modeling the additional concept of “negative dog,” or the negative activation of whatever neural topics correspond to the “dog” topic. Similarly, in a lie-detection experiment (Davatzikos et al., 2005), the goal is to model the presence or absence of top-down brain activity associated with deception, with no concept of “negative deception.” Furthermore, although the cognitive variables in these examples are binary, we do not restrict the activations of the topics to be strictly binary, since we don’t realistically expect neural topics to be activated in a

sudden all-or-none fashion when considered over the two second resolution of fMRI recordings.

The second motivation for the positive constraint is that positive topic activations greatly enhance the interpretability of the fitted results from the model. Using our interpretation of the neural topics parameters  $\mu_{k,v}$  as excitation and inhibition, allowing the activation of a neural topic  $z_{k,t}$  to be negative would change the sign of the topic's effect in the observed activity, effectively switching excitation to inhibition and inhibition to excitation. For example, negative activation of a neural topic corresponding to the processing of auditory stimuli, which might excite auditory cortex when positively activated, would then lead to *inhibition* of auditory cortex instead. Although it is unclear whether or not this scenario is neuroscientifically plausible, it is clear that interpreting the activation maps associated with a particular topic is much easier if one can easily distinguish inhibition from excitation.

3. *The fMRI pattern-classification problem is inherently over-specified, and regularization is required for any model to generalize properly.* As previously described in section 1.1, many fMRI datasets involve sparse observations of very high dimensional variables. Thus, the problem of fitting any sufficiently complex model to the data is *over-specified*: for any given instance of a dataset, there may be many solutions to the model that can explain the data equally well (Hastie et al., 2003). This is a problem for several reasons; with so many features and so few observations, there is a good chance that even with a relatively small number of reduced dimensions, we might expect the model to find an arbitrarily good fit to the observed data unless additional constraints are imposed. If we are attempting to generalize to a previously unobserved test set, the model will be at severe risk for over-fitting. Finally, if the solution to the model is not stable and unique, it is difficult (if not impossible) to draw reliable conclusions from experimental results.

One statistical approach to solving over-specified problems is to introduce *regularization*, which forces a model to minimize a penalty term on the parameters of the model in addition to an objective function used to assess the fit of the data. Common penalty functions are the L2 norm (used in ridge regression) or the L1 norm (known as “the Lasso” method in regression) (Hastie et al., 2003). The penalty functions force the model to use a “simpler” parameter set to explain the data than that which would be found by pure minimization of the objective function, where different penalty functions imply different definitions of “simple” parameter sets. Furthermore, regularization has been shown experimentally to be valuable when conducting ‘mind-reading’ type experiments. In a ‘brain activity interpretation’ competition held in 2006 (<http://www.ebc.pitt.edu/PBAIC.html>), in which teams competed to predict aspects of subject’s mental state while watching episodes of Home Improvement, the second place entry used ridge regression as part of their analysis<sup>1</sup> (Chigirev & Stephens, 2006). Regularization in the PACA model is accomplished through the use of prior distributions over the parameters; we will investigate later how these priors are equivalent to L2 and L1 regularizations in space and time, respectively (sections 2.4.2 and 2.4.3).

After we define these assumptions quantitatively, we shall see how these assumptions compare to the assumptions implicit in PCA and NMF. As it turns out, the “neural topics” we describe are essentially equivalent to the “principal components” that are found by PCA, but with the extra specification that the neural topics sum *additively*. Furthermore, we define our model *probabilistically* in terms of a generative random process, using prior probability distributions to add regularization to the fitting process, unlike PCA. This is the origin of the name for our new model, which we will analyze for the remainder of this chapter: “*Probabilistic Additive Component Analysis*.”

---

<sup>1</sup>It’s also worth noting that the third-place entry (Battle et al., 2006) used a graphical model as part of their analysis.

## 2.2 Model Definition

Now that we have motivated the PACA model, it is time to begin to define the model. At this point, it is useful to formalize the terms and notation we have introduced in the previous section.

### 2.2.1 Notation

We represent a given fMRI dataset as a  $T \times V$  observed matrix  $\mathbf{X}$ , where  $T$  is the number of recorded samples in the dataset and  $V$  is the number of voxels. The activity of the  $v$ 'th voxel at time  $t$  is notated  $x_{t,v}$ , and it is a scalar, real-valued number recorded by the scanner. Each row  $\mathbf{x}_t$  of the observed matrix  $\mathbf{X}$  is thus a pattern of activity across all recorded voxels in the brain at a single point in time, which we refer to as a *pattern*. Each column  $\mathbf{x}_{\langle \cdot \rangle, v}$  is a vector of observed values of a single voxel for the entire course of the experiment. In reality, each voxel  $v$  corresponds to a point in 3D Cartesian coordinate space, so there is a mapping  $f$  from the columns of  $\mathbf{X}$  to a 3D coordinate

$$f : \{1, \dots, V\} \rightarrow \{1, \dots, D\}^3,$$

where  $D$  is the dimension of a cube containing the subject's brain. However, although one might wish to construct a model that assumes relationships between neighboring voxels in 3D space (section 4.2), PACA (like PCA or NMF) indexes voxels through the arbitrary ordering induced by  $f$ . Similarly, although we describe  $\mathbf{X}$  as being indexed by time  $t$ , the actual ordering of the datapoints in the model is arbitrary (again, as in PCA or NMF).

We define a *neural topic* or just *topic* to be a row vector  $\boldsymbol{\mu}_k$  consisting of a single real valued *topic mean* for each voxel:

$$\boldsymbol{\mu}_k = \langle \mu_{k,0}, \mu_{k,1}, \dots, \mu_{k,V} \rangle.$$

If there are  $K$  different topics in our model, these are represented by a  $K \times V$  topic matrix  $\boldsymbol{\mu}$ . Each row of  $\boldsymbol{\mu}$  is a single neural topic, while each column  $\boldsymbol{\mu}_{\langle \cdot \rangle, v}$  is a vector of the  $K$  different topic means for the  $v$ 'th voxel in the dataset. At any time  $t$ , we define the *topic activation* to be a positive real number  $z_{k,t}$ ; the topic activation can be interpreted as the degree to which a given neural topic is influencing neural activity in the subject's brain.

Finally, we define the subject's *brain state* at time  $t$  to be a column vector  $\mathbf{z}_{\langle \cdot \rangle}$  consisting of the topic activations for each of the  $K$  topics at time  $t$ :

$$\mathbf{z}_{\langle \cdot \rangle, t} = \langle z_{0,t}, z_{1,t}, \dots, z_{K,t} \rangle.$$

The complete set of brain states for each point in time for a single dataset is then the  $K \times T$  matrix  $\mathbf{Z}$ . The columns of  $\mathbf{Z}$  are the brain states for each time-point as defined above, while each row  $\mathbf{z}_k$  is the activation of a single topic over the course of the entire experiment.

### 2.2.2 Generative Process

We can now formally define the generative model underlying PACA. We take a similar approach as in our definition of the coin flipping model described in section 1.3: we first define a generative process that incorporates our desired assumptions about the data, and then show how this process can be represented graphically.

In the PACA model, the observed data  $\mathbf{X}$  are generated through a three step random process (figure 2.3). The first step of the process is to generate the  $K$  neural topics for each voxel,  $\boldsymbol{\mu}$ . These are drawn randomly from a Normal distribution (figure 2.1) with mean  $\boldsymbol{\tau}$  and variance  $\sigma_\mu^2$ :

$$\mu_{k,v} \sim \mathcal{N}(\tau_{k,v}, \sigma_\mu^2), \quad \forall k, v. \tag{2.2}$$

Because each component of each topic is drawn independently, all topic means  $\mu_{k,v}$  will be independent of all other topic means given the parameters  $\tau_{k,v}$  and  $\sigma_m u^2$ . Thus, like the coin flips in the example in section 1.3, the individual voxels and topics are exchangeable.

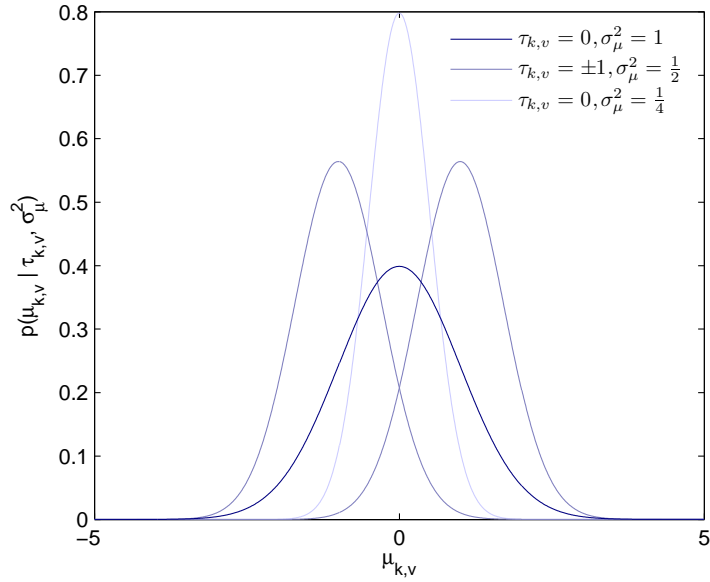


Figure 2.1: Example probability density functions of the Normal distribution, from which  $\boldsymbol{\mu}$  is drawn in the PACA model. The parameter  $\tau$  is the mean of the distribution, while  $\sigma_\mu^2$  is the variance. The distribution is plotted for several values of  $(\tau, \sigma_\mu^2)$ .

The Normal distribution was chosen as the prior for  $\boldsymbol{\mu}$  to satisfy our first neuroscientific assumption; because the domain of the Normal density function is the real line, the values of each  $\mu_{k,v}$  are unconstrained. Furthermore, by placing a prior distribution on  $\boldsymbol{\mu}$ , we will see that we satisfy our third motivating principle, as the  $\sigma_\mu^2$  parameter enforces an L2 norm regularization of the topic means (section 2.4.2).

The next step of the process is to generate the brain states for each point in time,  $\mathbf{Z}$ . These are drawn from a Gamma distribution (figure 2.2) with shape  $a$  and scale  $b$ :

$$z_{k,t} \sim \text{Gamma}(a, b). \quad (2.3)$$

Again, each component of each brain state is drawn independently from the same distribution, so the time-points are considered exchangeable. The Gamma distribution was chosen to satisfy the second motivating assumption behind the model: since the domain of the Gamma density function is over the positive reals,  $\mathbf{Z}$  is constrained to be non-negative.

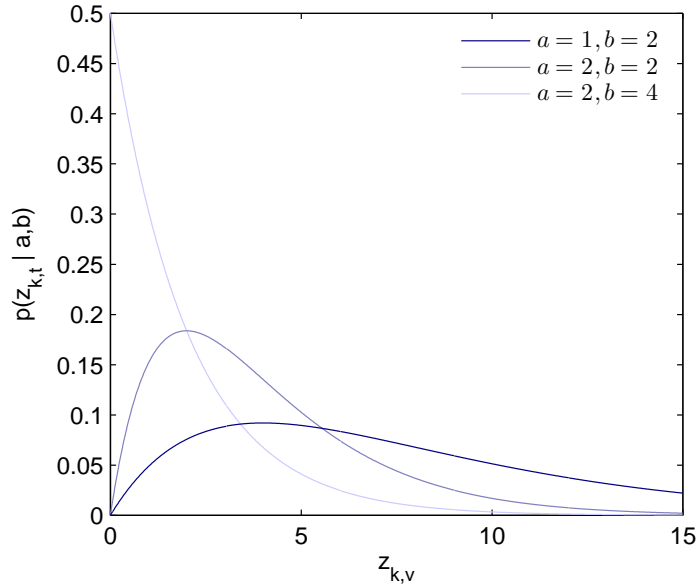


Figure 2.2: Example probability density functions of the Gamma distribution, from which  $\mathbf{Z}$  is drawn in the PACA model. The parameter  $a$  is the *shape* of the distribution, while  $b$  is the *scale*. The distribution is plotted for several values of  $(a, b)$ .

Furthermore, as we will see in section 2.4.3, the Gamma prior enforces an L1 norm regularization over the brain states with certain parameter settings, again satisfying the third motivating assumption for PACA.

The third and final step of the generative process is to generate the observed data matrix  $\mathbf{X}$ . The activation of voxel  $v$  at time  $t$  is assumed to be drawn from a Normal distribution where the mean is the additive summation of the topic means  $\mu_{k,v}$  weighted by the topic activations  $z_{k,v}$ , and the variance is defined by a parameter  $\sigma_x^2$ . Thus, we define the conditional probability of  $x_{t,v}$  given the corresponding topic vectors  $\mu_{\langle \cdot \rangle, v}$  and  $z_{\langle \cdot \rangle, t}$ :

$$x_{t,v} | \mu_{\langle \cdot \rangle, v}, z_{\langle \cdot \rangle, t} \sim \mathcal{N}(\hat{x}_{t,v}, \sigma_x^2), \quad (2.4)$$

where

$$\hat{x}_{t,v} = \sum_k z_{k,t} \mu_{k,v}. \quad (2.5)$$

Equivalently, we can define  $x_{t,v}$  exactly as in equation 2.1 by setting the noise term  $\epsilon$  to be

---

**Figure 2.3** The generative process formulation of PACA.

---

Choose fixed values for hyperparameters  $a, b, \sigma_\mu^2, \sigma_x^2$ , and  $\tau$ .

1. Draw  $\mu_k \sim \mathcal{N}(\tau_k, \sigma_\mu^2 \mathbf{I})$  for all  $k \in \{1, \dots, K\}$ .
  2. For each time-point  $t \in \{1, \dots, T\}$ :
    - (a) Draw  $z_{k,t} \sim \text{Gamma}(a, b)$  for all  $k \in \{1, \dots, K\}$ .
    - (b) For each voxel  $v \in \{1, \dots, V\}$ :
      - i. Compute the predicted mean  $\hat{x}_{t,v} = \sum_{k=1}^K \mu_{k,v} z_{k,t}$  for voxel  $v$  at time  $t$ .
      - ii. Draw  $x_{t,v} \sim \mathcal{N}(\hat{x}_{t,v}, \sigma_x^2)$ .
- 

normally distributed with mean zero and variance  $\sigma_x^2$ . Thus, the parameter  $\sigma_x^2$  can be interpreted as the variance of the noise in the observed data, and the sum  $\hat{x}_{t,v}$  can be interpreted as the predicted activation of voxel  $v$  at time  $t$  in the model.

We can now summarize the various random variables, parameters, and hyperparameters of the PACA model. In this model, the only observed quantity is the matrix  $\mathbf{X}$ , which we have assumed has noise with variance  $\sigma_x^2$ . The latent variables of the model are  $\mu$  and  $\mathbf{Z}$ , which are also the parameters of  $\mathbf{X}$  in the conditional probability distribution  $p(\mathbf{X}|\mu, \mathbf{Z})$ . The latent variables  $\mathbf{Z}$  and  $\mu$  are in turn parameterized by the fixed sets of hyperparameters  $\{a, b\}$  and  $\{\tau, \sigma_\mu^2\}$ . The model is also defined by the dimensions of the variables:  $V, T$ , and  $K$ . All of the constants, variables, and parameters are summarized for reference in table 2.1. Finally, we note that we have included all three of our primary assumptions into the PACA model: the matrix  $\mu$  consists of  $K$  neural “topics” that sum additively according to the topic activations in the matrix  $\mathbf{Z}$  to form the observed data, and both  $\mu$  and  $\mathbf{Z}$  have prior distributions that will enforce regularization when fitting.

### 2.2.3 Probabilistic Graphical Model and Full Joint Distribution

From the generative process summarized in figure 2.3, it is easy to draw an equivalent graphical model. Following the procedure outlined in section 1.3, we build a graph with

Symbol	Type	Dim	Prior	Description
$\mathbf{X}$	Observed r.v.	$T \times V$	Normal	Observed brain patterns.
$\boldsymbol{\mu}$	Latent r.v.	$K \times V$	Normal	Topic means.
$\mathbf{Z}$	Latent r.v.	$K \times T$	Gamma	Brain states.
$\sigma_x^2$	Hyperparameter	Scalar	None	Variance of all $X_{t,v}$
$\sigma_\mu^2$	Hyperparameter	Scalar	None	Variance of all $\mu_{k,v}$
$\boldsymbol{\tau}$	Hyperparameter	$K \times V$	None	Mean for each $\mu_{k,v}$
$a, b$	Hyperparameter	Scalar	None	Parameters for the Gamma $Z$ prior

Table 2.1: The variables (latent and observed) and hyperparameters of the PACA generative model.

nodes for the random variables  $\boldsymbol{\mu}$ ,  $\mathbf{Z}$ ,  $\mathbf{X}$ , and the hyperparameters  $a$ ,  $b$ ,  $\sigma_\mu^2$ , and  $\sigma_x^2$ .  $\mathbf{X}$  depends on both  $\boldsymbol{\mu}$  and  $\mathbf{Z}$ , in addition to  $\sigma_x^2$ , so edges are added from  $\boldsymbol{\mu}$  to  $\mathbf{X}$ , from  $\mathbf{Z}$  to  $\mathbf{X}$ , and from  $\sigma_x^2$  to  $\mathbf{X}$ . However,  $\boldsymbol{\mu}$  depends on  $\boldsymbol{\tau}$  and  $\sigma_\mu^2$ , and  $\mathbf{Z}$  depends on  $a$  and  $b$ , so these edges are needed as well. Finally, only  $\mathbf{X}$  is observed, and  $a$ ,  $b$ ,  $\sigma_\mu^2$  and  $\sigma_x^2$  are hyperparameters that do not change, so we decorate the nodes accordingly. The completed graph is shown in figure 2.4.

From the graph in figure 2.4 and the conditional probability distributions we defined when expressing PACA as a random process (figure 2.3), it is now easy to obtain the full joint distribution of the PACA model. Using the formula in equation 1.2, we write the joint distribution as the product of the conditional probability distribution of every node in the graph, given its parents. Thus, the joint probability of observations of the random matrices  $\boldsymbol{\mu}$ ,  $\mathbf{Z}$ , and  $\mathbf{X}$  given the set of fixed hyperparameters  $\eta = \{\sigma_x^2, \sigma_\mu^2, \boldsymbol{\tau}, a, b\}$  can be written as following:

$$p(\mathbf{X}, \boldsymbol{\mu}, \mathbf{Z} | \eta) = p(\mathbf{Z} | a, b) p(\boldsymbol{\mu} | \boldsymbol{\tau}, \sigma_\mu^2) p(\mathbf{X} | \boldsymbol{\mu}, \mathbf{Z}, \sigma_x^2), \quad (2.6)$$

where

$$p(\mathbf{Z} | a, b) = \prod_k \prod_t z_{k,t}^{a-1} \frac{1}{b^a \Gamma(a)} \exp \{-z_{k,t}/b\}, \quad (2.7)$$

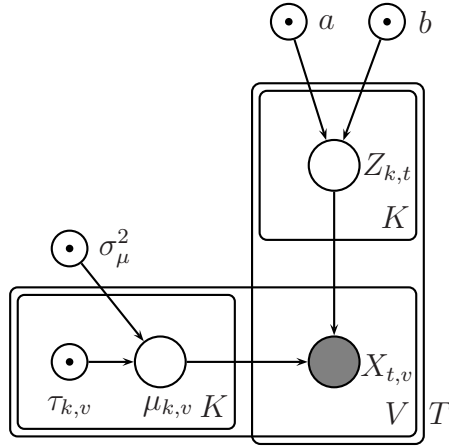


Figure 2.4: The probabilistic graphical model representation of Probabilistic Additive Components Analysis (PACA). Each observed data point  $x_{t,v}$  is Normally distributed around the weighted additive sum of the  $K$  latent components  $\mu_{k,v}$  of voxel  $v$  according to the  $K$  non-negative coefficients  $z_{k,t}$  for time  $t$ ,  $z_{k,t}$ , with variance  $\sigma_x^2$ . The unobserved additive components or “neural topics”  $\mu_{k,v}$  are normally distributed with mean  $\tau_{k,v}$  and variance  $\sigma_\mu^2$ , and the non-negative coefficients or “brain states”  $z_{k,t}$  are distributed according to a Gamma distribution with shape  $a$  and scale  $b$ .

as defined by the probability density function of a Gamma distribution,

$$p(\boldsymbol{\mu}|\boldsymbol{\tau}, \sigma_\mu^2) = \prod_k \prod_v \frac{1}{(2\pi\sigma_\mu^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma_\mu^2} (\mu_{k,v} - \tau_{k,v})^2 \right\}, \quad (2.8)$$

as defined by the probability density function of a Normal distribution, and

$$p(\mathbf{X}|\boldsymbol{\mu}, \mathbf{Z}, \sigma_x^2) = \prod_t \prod_v \frac{1}{(2\pi\sigma_x^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma_x^2} (x_{t,v} - \hat{x}_{t,v})^2 \right\}. \quad (2.9)$$

again, applying the probability density function of a Normal distribution and using the definition of  $\hat{x}_{t,v}$  defined in equation 2.5. Thus, the probability of observing  $\mathbf{X}$ ,  $\boldsymbol{\mu}$ , and  $\mathbf{Z}$  together is proportional to the probability of observing  $\boldsymbol{\mu}$  and  $\mathbf{Z}$  individually and the probability of generating the observed values of  $\mathbf{X}$  from  $\boldsymbol{\mu}$  and  $\mathbf{Z}$  with noise governed by  $\sigma_x^2$ .

## 2.3 Estimating $\boldsymbol{\mu}$ and $\mathbf{Z}$ from Data

Now that we have fully specified all relevant theoretical aspects of the model, we show how the latent variables  $\boldsymbol{\mu}$  and  $\mathbf{Z}$  can be estimated from a given set of observed data  $\mathbf{X}$ . We use the method of *maximum a posteriori* estimation introduced in section 1.4; we will see in the next section that this method for estimating the latent variables can be used to characterize PACA purely in terms of a regularized least-squared-error matrix factorization, without an explicit probabilistic model.

We now apply the method of MAP estimation to the PACA model in this paper. Although  $\boldsymbol{\mu}$  and  $\mathbf{Z}$  are latent random variables, we define the parameters  $\theta$  to be

$$\theta = \{\boldsymbol{\mu}, \mathbf{Z}\}, \quad (2.10)$$

since  $\boldsymbol{\mu}$  and  $\mathbf{Z}$  are the parameters for the modeled distribution of  $\mathbf{X}$ , and these are the variables which we want to estimate. Following the formulation given in section 1.4,  $a$ ,  $b$ ,  $\sigma_\mu^2$ ,  $\boldsymbol{\tau}$ , and  $\sigma_x^2$  are the hyperparameters of the model:

$$\eta = \{\boldsymbol{\tau}, a, b, \sigma_\mu^2, \sigma_x^2\}. \quad (2.11)$$

The hyperparameters are considered fixed, along with  $\mathbf{X}$  for the MAP maximization problem. The observed dataset  $\mathcal{D}$  is then simply  $\mathbf{X}$ . We can then write the log posterior (equation 1.12) of the PACA model in terms of the log likelihoods of the data and the parameters as follows:

$$\mathcal{L}(\theta; \mathcal{D}, \eta) = L(\theta; \mathcal{D}) + L(\eta; \theta) \quad (2.12)$$

$$= L(\boldsymbol{\mu}, \mathbf{Z}; \mathbf{X}) + L(\boldsymbol{\tau}, \sigma_\mu^2; \boldsymbol{\mu}) + L(a, b; \mathbf{Z}), \quad (2.13)$$

where

$$L(\boldsymbol{\mu}, \mathbf{Z}; \mathbf{X}) = -\frac{TV}{2} \log(2\pi\sigma_x^2) - \frac{1}{2\sigma_x^2} \sum_t \sum_v (x_{t,v} - \hat{x}_{t,v})^2, \quad (2.14)$$

$$L(\boldsymbol{\tau}, \sigma_\mu^2; \boldsymbol{\mu}) = -\frac{KV}{2} \log(2\pi\sigma_\mu^2) - \frac{1}{2\sigma_\mu^2} \sum_k \sum_v (\mu_{k,v} - \tau_{k,v})^2, \quad (2.15)$$

and

$$L(a, b; \mathbf{Z}) = \sum_k \left[ \sum_t \left[ (a-1) \log z_{k,t} - \frac{z_{k,t}}{b} \right] - T \log \Gamma(a) - Ta \log b \right], \quad (2.16)$$

are the individual log likelihoods of  $\mathbf{X}$ ,  $\boldsymbol{\mu}$ , and  $\mathbf{Z}$ , respectively, and  $\hat{x}_{t,v}$  is defined as in equation 2.5 The MAP estimation problem for PACA is therefore

$$\hat{\theta}_{\text{MAP}} = \underset{\boldsymbol{\mu}, \mathbf{Z}}{\text{argmax}} [L(\boldsymbol{\mu}, \mathbf{Z}; \mathbf{X}) + L(\boldsymbol{\tau}, \sigma_\mu^2; \boldsymbol{\mu}) + L(a, b; \mathbf{Z})], \quad (2.17)$$

subject to the constraint

$$\mathbf{Z} \geq 0. \quad (2.18)$$

We can attempt to solve this problem by taking the derivative of the log posterior with respect to  $\boldsymbol{\mu}$  and  $\mathbf{Z}$ :

$$\frac{\partial \mathcal{L}}{\partial z_{k_0, t_0}} = \frac{a-1}{z_{k_0, t_0}} - \frac{1}{b} + \frac{1}{\sigma_x^2} \sum_v [(x_{t_0, v} - \hat{x}_{t_0, v}) \mu_{k_0, v}], \quad (2.19)$$

and

$$\frac{\partial \mathcal{L}}{\partial \mu_{k_0, v_0}} = -\frac{1}{\sigma_\mu^2} (\mu_{k_0, v_0} - \tau_{k_0, v_0}) + \frac{1}{\sigma_x^2} \sum_t [(x_{t, v_0} - \hat{x}_{t, v_0}) z_{k_0, t}]. \quad (2.20)$$

Unfortunately, these derivatives cannot be analytically solved to find the critical points of the log posterior function. Therefore, we must use numerical methods to find the MAP estimators; in this thesis, we use a conjugate gradient descent algorithm to solve equation 2.17. However, optimizing under the inequality constraint in equation 2.18 poses a number of practical challenges to this procedure. Thankfully, we can sidestep these issues entirely

through a simple reparametrization of  $\mathbf{Z}$  that converts the constrained maximization in equation 2.17 into an unconstrained one.

### 2.3.1 Unconstrained Optimization

To convert the constrained maximization problem in equation 2.17 to an unconstrained maximization problem, we apply the following reparameterization of  $\mathbf{Z}$ :

$$z_{k,t} = \exp\{w_{k,t}\}, \quad \forall k, v \quad (2.21)$$

where  $w_{k,t}$  is unconstrained. Since exponentiation is a monotonically increasing function, the maximum of  $\mathcal{L}$  with respect to  $\mathbf{W}$  will be equal to the maximum with respect  $\mathbf{Z}$ , and given  $\widehat{\mathbf{W}}_{\text{MAP}}$ , we can easily find  $\widehat{\mathbf{Z}}_{\text{MAP}}$ . However, because exponentiation maps the real line onto the positive reals, the maximization problem with respect to  $\mathbf{W}$  is now unconstrained.

Plugging in  $\mathbf{W}$  for  $\mathbf{Z}$  results in the following set of changed log likelihood functions:

$$L(a, b; \mathbf{W}) = \sum_k \left[ \sum_t \left[ (a-1)w_{k,t} - \frac{e^{w_{k,t}}}{b} \right] - T \log \Gamma(a) - Ta \log b \right], \quad (2.22)$$

and

$$L(\boldsymbol{\mu}, \mathbf{W}; \mathbf{X}) = -\frac{TV}{2} \log(2\pi\sigma_x^2) - \frac{1}{2\sigma_x^2} \sum_t \sum_v \left[ x_{t,v} - \left( \sum_k \mu_{k,v} e^{w_{k,t}} \right) \right]^2. \quad (2.23)$$

The new derivatives are as follows:

$$\frac{\partial \mathcal{L}}{\partial w_{k_0, t_0}} = (a-1) - \frac{e^{w_{k_0, t_0}}}{b} + \frac{1}{\sigma_x^2} \sum_v \left[ \left( x_{t_0, v} - \sum_k \mu_{k, v} e^{w_{k, t_0}} \right) \mu_{k_0, v} e^{w_{k_0, t_0}} \right], \quad (2.24)$$

and

$$\frac{\partial \mathcal{L}}{\partial \mu_{k_0, v_0}} = -\frac{1}{\sigma_\mu^2} (\mu_{k_0, v_0} - \tau_{k, v}) + \frac{1}{\sigma_x^2} \sum_t \left[ \left( x_{t, v_0} - \sum_k \mu_{k, v_0} e^{w_{k, t}} \right) e^{w_{k_0, t}} \right]. \quad (2.25)$$

Thus, for a given set of hyperparameters  $a$ ,  $b$ ,  $\tau$ ,  $\sigma_\mu^2$ , and  $\sigma_x^2$ , and an observed dataset  $\mathbf{X}$ , we can find the MAP estimators  $\hat{\boldsymbol{\mu}}_{\text{MAP}}$  and  $\hat{\mathbf{Z}}_{\text{MAP}}$  by directly applying methods for unconstrained optimization.

## 2.4 Setting Hyperparameters

So far, we have not yet dealt with how to set values for the hyperparameters,  $a$ ,  $b$ ,  $\tau$ ,  $\sigma_\mu^2$ , and  $\sigma_x^2$ . We now present a principled method of selecting hyperparameters based on the interpretation of the prior distributions as regularizations of the neural topics and brain states, respectively.

Let us start by considering the dual minimization problem to the maximization problem in equation 2.17:

$$\{\hat{\boldsymbol{\mu}}_{\text{MAP}}, \hat{\mathbf{Z}}_{\text{MAP}}\} = \underset{\boldsymbol{\mu}, \mathbf{Z}}{\operatorname{argmin}} [-L(\boldsymbol{\mu}, \mathbf{Z}; \mathbf{X}) - L(\tau, \sigma_\mu^2; \boldsymbol{\mu}) - L(a, b; \mathbf{Z})]. \quad (2.26)$$

The formulation in equation 2.26 is simply minimizing the negative log posterior function, but at this point, we can ignore the underlying probabilistic context and consider equation 2.26 purely from the perspective of an optimization problem. Because we are minimizing, we can remove all constant terms from the various terms of the equation to simplify the analysis. Furthermore, we can analyze each term in equation 2.26 separately to determine the effect of the hyperparameters on that part of the minimization problem. Once we understand the effect of each hyperparameter on the fit of the data, we can determine which hyperparameters should be tweaked when fitting the model, and how the values should be set.

### 2.4.1 Minimization of Squared Reconstruction Error

We start by considering the first term in equation 2.26, the negative log likelihood of the observed data  $\mathbf{X}$  given the latent variables and the hyperparameter  $\sigma_x^2$ . We start by plugging in equation 2.14 into the optimization problem of equation 2.26, but without the other terms:

$$\{\boldsymbol{\mu}^*, \mathbf{Z}^*\} = \underset{\boldsymbol{\mu}, \mathbf{Z}}{\operatorname{argmin}} \left[ \frac{TV}{2} \log(2\pi\sigma_x^2) + \frac{1}{2\sigma_x^2} \sum_t \sum_v (x_{t,v} - \hat{x}_{t,v})^2 \right] \quad (2.27)$$

$$= \underset{\boldsymbol{\mu}, \mathbf{Z}}{\operatorname{argmin}} \left[ \frac{1}{2\sigma_x^2} \sum_t \sum_v (x_{t,v} - \hat{x}_{t,v})^2 \right]. \quad (2.28)$$

Now, we note that the predicted voxel means  $\hat{x}_{t,v}$  can be written equivalently as the dot product of the  $t$ 'th column of  $\mathbf{Z}$  with the  $v$ 'th column of  $\boldsymbol{\mu}$ :

$$\hat{x}_{t,v} = \sum_k z_{k,t} \mu_{k,v} = \mathbf{z}_{(\cdot),t}^T \cdot \boldsymbol{\mu}_{(\cdot),v}.$$

Thus, the entire  $T \times V$  matrix of predicted voxel means  $\widehat{\mathbf{X}}$  can be written

$$\widehat{\mathbf{X}} = \mathbf{Z}^T \boldsymbol{\mu}, \quad (2.29)$$

where  $\widehat{\mathbf{X}}$  is called the *reconstruction* of the original data  $\mathbf{X}$  from the latent variables  $\boldsymbol{\mu}$  and  $\mathbf{Z}$ . The sum of squared distances between  $\mathbf{X}$  and  $\widehat{\mathbf{X}}$  is thus the *reconstruction error*, and it can be written,

$$\|\mathbf{X} - \widehat{\mathbf{X}}\|_2^2 = \sum_t \sum_v (x_{t,v} - \hat{x}_{t,v})^2. \quad (2.30)$$

Thus, we see by comparing equations 2.30 and 2.28 that minimizing the negative log likelihood of the data is equivalent to minimizing the reconstruction error of  $\boldsymbol{\mu}$  and  $\mathbf{Z}$ ,

$$\{\boldsymbol{\mu}^*, \mathbf{Z}^*\} = \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \left[ \frac{1}{2\sigma_x^2} \sum_t \sum_v (x_{t,v} - \hat{x}_{t,v})^2 \right] \quad (2.31)$$

$$= \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \frac{1}{\sigma_x^2} \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2, \quad (2.32)$$

where the reconstruction error is scaled by the inverse variance of the error  $\frac{1}{\sigma_x^2}$ . To simplify our analysis, we can set  $\sigma_x^2 = 1$  without changing the minimum of the reconstruction error function.

## 2.4.2 L2 Regularization of Topic Means

We now consider the second term in equation 2.26. Let  $\tau_{k,v} = 0$ , for all  $k, v$ . Then minimizing the negative log likelihood of the latent variable  $\boldsymbol{\mu}$  (equation 2.15) becomes

$$\{\boldsymbol{\mu}^*, \mathbf{Z}^*\} = \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \left[ \frac{KV}{2} \log(2\pi\sigma_\mu^2) + \frac{1}{2\sigma_\mu^2} \sum_k \sum_v (\mu_{k,v} - \tau_{k,v})^2 \right] \quad (2.33)$$

$$= \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \left[ \frac{1}{2\sigma_\mu^2} \sum_k \sum_v \mu_{k,v}^2 \right] \quad (2.34)$$

$$= \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \frac{1}{2\sigma_\mu^2} \|\boldsymbol{\mu}\|_2^2. \quad (2.35)$$

Thus, when  $\tau_{k,v} = 0$ , the Gaussian prior over  $\mu$  is exactly equivalent to an L2 regularization over each neural topic. Therefore, when the PACA model is fit using MAP estimation in equation 2.26, the model must balance the minimization of the reconstruction error with the minimization of the topic means themselves. The balance is controlled by the variance of the  $\boldsymbol{\mu}$  prior,  $\sigma_\mu^2$ . This sort of L2 norm based regularization is used in the ‘‘ridge regression’’ analysis technique, and it is widely known that L2 regularization will result in a smooth distribution over the regularized parameters (Hastie et al., 2003). If  $\tau_{k,v}$  was not zero, the regularization would serve to bring  $\boldsymbol{\mu}$  closer to  $\boldsymbol{\tau}$ , potentially accounting for whitening in

the data. For our purposes, since all the data is z-scored,  $\tau_{k,v} = 0$  is sufficient.

### 2.4.3 L1 Regularization of Brain States

Finally, we examine the last term of equation 2.26. Consider the case where  $a = 1$ . Then the minimization becomes

$$\begin{aligned} \{\boldsymbol{\mu}^*, \mathbf{Z}^*\} &= \underset{\boldsymbol{\mu}, \mathbf{Z}}{\operatorname{argmin}} \sum_k \sum_t - \left[ (a-1) \log z_{k,t} - \frac{z_{k,t}}{b} \right] + \\ &\quad \sum_k [T \log \Gamma(a) - Ta \log b] \end{aligned} \quad (2.36)$$

$$= \underset{\boldsymbol{\mu}, \mathbf{Z}}{\operatorname{argmin}} \sum_k \sum_t \frac{z_{k,t}}{b} \quad (2.37)$$

$$= \underset{\boldsymbol{\mu}, \mathbf{Z}}{\operatorname{argmin}} \frac{1}{b} \|\mathbf{Z}\|_1. \quad (2.38)$$

Thus, when  $a = 1$ , minimizing the negative log likelihood of  $\mathbf{Z}$  is equivalent to minimizing the L1 norm of  $\mathbf{Z}$  (since  $\mathbf{Z}$  is non-negative, so that  $\mathbf{Z} = |\mathbf{Z}|$ ), scaled by the inverse of hyperparameter  $b$ . This is exactly equivalent to L1 regularization, which is known in linear regression statistics as the “lasso” method (Hastie et al., 2003).

However, it is also well known that L1 regularization will cause the fitting algorithm to find a sparse distribution over the regularized parameters (Hastie et al., 2003). Since we have already described reasons for why very sparse brain states are not necessarily desirable (section 2.1), we consider the case where  $a = 1 + b^{-1}$ , so that  $a > 1$ . In this case,

the minimization becomes

$$\begin{aligned} \{\boldsymbol{\mu}^*, \mathbf{Z}^*\} &= \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \sum_k \sum_t - \left[ \frac{1}{b} \log z_{k,t} - \frac{z_{k,t}}{b} \right] + \\ &\quad \sum_k \left[ T \log \Gamma\left(\frac{1}{b} + 1\right) - T\left(\frac{1}{b} + 1\right) \log b \right] \end{aligned} \quad (2.39)$$

$$= \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \frac{1}{b} \sum_k \sum_t z_{k,t} - \log z_{k,t} \quad (2.40)$$

$$= \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \frac{1}{b} (\|\mathbf{Z}\|_1 - \sum_{k,t} \log z_{k,t}) \quad (2.41)$$

$$= \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \frac{1}{b} (\|\mathbf{Z}\|_1 - \|\mathbf{Z}\|_{\log}) \quad (2.42)$$

Thus, in the case where  $a > 1$ , the Gamma prior over  $\mathbf{Z}$  is equivalent to an L1 regularization with the extra term  $\sum_{k,t} \log z_{k,t}$  (we which notate as  $\|\mathbf{Z}\|_{\log}$ ) subtracted. As any  $z_{k,t}$  approaches zero,  $\log z_{k,t}$  will approach negative infinity, so subtracting the log of each  $z_{k,t}$  will prevent any  $z_{k,t}$  from becoming exactly zero. (In general, the negative log function can be used *barrier* function that will prevent absolute zero values in minimization problems.) By restricting our analysis to the case where  $a > 1$ , we can ensure that the distributions over  $\mathbf{Z}$  are not extremely sparse. We can also arrive at the same conclusion by examining the probability density of the Gamma distribution with different parameter sets (figure 2.2); for the case where  $a > 1$ , the peak of the density function is positive and shifted away from zero.

#### 2.4.4 Alternative Formulation: Regularized Matrix Factorization

One useful consequence of the previous results is that we can re-express the probabilistic model of PACA as a matrix factorization problem. Combining the minimization problems in equations 2.35, 2.42, and 2.28 back into the MAP estimation problem, we can express the parameter estimation process originally derived from the probabilistic model as the

solution to the regularization matrix factorization problem,

$$\{\boldsymbol{\mu}^*, \mathbf{Z}^*\} = \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \frac{1}{2} \|\mathbf{X} - \mathbf{Z}^T \boldsymbol{\mu}\|_2^2 + \frac{1}{2\sigma_\mu^2} \|\boldsymbol{\mu}\|_2^2 + \frac{1}{b} (\|\mathbf{Z}\|_1 - \|\mathbf{Z}\|_{\log}), \quad (2.43)$$

subject to the constraint  $\mathbf{Z} \geq 0$ . We will see that both PCA and NMF can be expressed as similar matrix factorization problems, but with different constraints and no regularization terms (table 2.2).

### 2.4.5 Setting hyperparameters for Proportional Regularization

From the matrix factorization of equation 2.43, we see that the hyperparameters of our probability model scale the amount of regularization in the corresponding matrix factorization problem. We can therefore set the hyperparameters to enforce a desired level of regularization in the fit of the model to the observed data. However, for a constant set of hyperparameters, the impact of the regularization terms will decrease as the dimensionality of the observed data increases. To see why, consider the case where  $\mathbf{X}$ ,  $\boldsymbol{\mu}$ , and  $\mathbf{Z}$  have very low variance, so that each observation  $x_{t,v}$ ,  $\mu_{k,v}$ , and  $z_{k,t}$  are close to their means  $\bar{x}$ ,  $\bar{\mu}$ , and  $\bar{z}$ . Equation 2.43 then becomes

$$\{\boldsymbol{\mu}^*, \mathbf{Z}^*\} = \operatorname{argmin}_{\boldsymbol{\mu}, \mathbf{Z}} \frac{TV}{2} \|\bar{x} - K\bar{\mu}\bar{z}\|^2 + \frac{KV}{\sigma_\mu^2} \bar{\mu}^2 + \frac{KT}{b} (\bar{z} - \log \bar{z}). \quad (2.44)$$

Because there are  $TV$  error terms, the squared error is scaled by  $TV$ , but there are only  $KV$  regularization terms for  $\boldsymbol{\mu}$  and only  $KT$  regularization terms for  $\mathbf{Z}$ . In our fMRI datasets,  $V \gg T \gg K$ , so the  $TV$  squared error terms will dominate the minimization as the dimensionality of the observed data increases. Similarly, if the number of reduced dimensions  $K$  increases, the importance of the regularization terms will be increased.

If we desire to apply a constant amount of regularization to the matrix factorization regardless of the dimensionality of the data, then as the dimensions of the data increase, we will need to adjust our hyperparameters proportionately. In general, the decreasing

importance of the prior distribution as more data is collected is in accord with a purely Bayesian approach to model fitting. However, we wish to take a more pragmatic approach that will allow us to set the hyperparameters in a principled fashion for datasets of different dimensions. We do this by reparameterizing the hyperparameters  $a$ ,  $b$ , and  $\sigma_\mu^2$  so that they are scaled with the dimensions of the data.

Consider now a reparameterization of the hyperparameters,  $\lambda$  and  $\gamma$ :

$$\sigma_\mu^2 = \frac{K}{T\lambda}, \quad b = \frac{2K}{V\gamma}, \quad a = b^{-1} + 1. \quad (2.45)$$

Equation 2.44 now becomes

$$\{\mathbf{Z}^*, \boldsymbol{\mu}^*\} = \operatorname{argmin}_{\mathbf{Z}, \boldsymbol{\mu}} \frac{TV}{2} \|\bar{x} - \hat{x}\|^2 + \frac{KV\lambda T}{2K} \bar{\mu}^2 + \frac{KVT\gamma}{2K} (\bar{z} - \log \bar{z}) \quad (2.46)$$

$$= \operatorname{argmin}_{\mathbf{Z}, \boldsymbol{\mu}} \frac{TV}{2} (\|\bar{x} - \hat{x}\|^2 + \lambda \bar{\mu}^2 + \gamma (\bar{z} - \log \bar{z})) \quad (2.47)$$

$$= \operatorname{argmin}_{\mathbf{Z}, \boldsymbol{\mu}} \|\bar{x} - \hat{x}\|^2 + \lambda \bar{\mu}^2 + \gamma (\bar{z} - \log \bar{z}), \quad (2.48)$$

so the reparameterization factors out the dimensionality of the data from the minimization problem. Thus, using  $\lambda$  and  $\gamma$ , we can explicitly control the balance of regularization and error minimization.

Returning to the matrix factorization problem, we plug  $\lambda$  and  $\gamma$  back into equation 2.43.

The final version of the matrix factorization is thus

$$\{\mathbf{Z}^*, \boldsymbol{\mu}^*\} = \operatorname{argmin}_{\mathbf{Z}, \boldsymbol{\mu}} \frac{1}{2} \|\mathbf{X} - \boldsymbol{\mu}^T \mathbf{Z}\|_2^2 + \lambda \frac{T}{2K} \|\boldsymbol{\mu}\|_2^2 + \gamma \frac{V}{2K} (\|\mathbf{Z}\|_1 - \|\mathbf{Z}\|_{\log}). \quad (2.49)$$

Although the version of PACA implemented in this thesis uses the probabilistic version of the model to find MAP estimates for  $\boldsymbol{\mu}$  and  $\mathbf{Z}$ , we choose settings for  $a$ ,  $b$ , and  $\sigma_\mu^2$  based on the parameterizations in equation 2.49. To ensure that equation 2.43 holds, and because we use centered data  $\mathbf{X}$  with mean 0, we set  $\boldsymbol{\tau} = 0$ . Finally, given  $\lambda$  and  $\gamma$  in equation 2.43, there is no need for a third coefficient to scale the squared error, so we use  $\sigma_x^2 = 1$ . This

Method	Minimization	Constraints
PACA	$\ \mathbf{X} - \mathbf{Z}^T \boldsymbol{\mu}\ _2^2 + \lambda \ \boldsymbol{\mu}\ _2^2 + \gamma (\ \mathbf{Z}\ _1 - \ \mathbf{Z}\ _{\log})$	$\mathbf{Z} \geq 0$
NMF <sup>3</sup>	$\ \mathbf{X} - \mathbf{Z}^T \boldsymbol{\mu}\ _2^2$	$\boldsymbol{\mu}, \mathbf{Z} \geq 0$
PCA <sup>4</sup>	$\ \mathbf{X} - \mathbf{Z}^T \boldsymbol{\mu}\ _2^2$	None

Table 2.2: PACA, NMF, and PCA compared, when they are expressed in terms of matrix factorizations.

method of selecting parameters was used in all analyses unless otherwise noted.

## 2.5 Comparison to Existing Methods

### 2.5.1 Principal Component Analysis (PCA)

Principal Component Analysis, in contrast to our model, is predicated on the goal of finding a set of  $K$  uncorrelated dimensions of a given  $N \times P$  dataset  $\mathbf{X}$  such that variance of the reduce dataset is maximized. These dimensions are known as the *principal components*, and it can be shown that the principal components of a given dataset  $\mathbf{X}$  are the eigenvectors of the sample correlation matrix  $\Sigma$  (Jolliffe, 2002). The variance of each principal component is given by the corresponding eigenvalue. Thus, the first principal component, which explains the most variance in the original data, is the eigenvector with the largest eigenvalue, the second principal component is the eigenvector with the second largest, and so forth.

Although it has been claimed that PCA has no explicit model (Jolliffe, 2002), Tipping and Bishop (1999) presented a probabilistic version of PCA with an associated graphical model whose maximum likelihood estimators are equivalent to a projection on the *principal subspace* of the data defined by the principal components (Tipping & Bishop, 1999). Thus, we can directly compare the probabilistic assumptions underlying PCA with those of our model.

<sup>3</sup>This formulation of non-negative matrix factorization is discussed in (Lee & Seung, 2001).

<sup>4</sup>The minimizing least-squares formulation of PCA is presented in (Hastie et al., 2003).

In the probabilistic PCA (PPCA) model, as in our model, the observed datapoints  $\mathbf{x}$  are assumed to be normally distributed:

$$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}|\mathbf{W}\mathbf{z} + \boldsymbol{\tau}, \sigma^2\mathbf{I}), \quad (2.50)$$

where  $\mathbf{z}$  is a matrix of normally distributed latent variables,  $\mathbf{W}$  is a matrix relating the latent variables to the observed variables, and  $\boldsymbol{\tau}$  is a matrix of parameters allowing the model to have non-zero mean (Tipping & Bishop, 1999). Thus, like our model, each  $x_{t,v}$  can be expressed as a linear combination of latent variables (ignoring  $\boldsymbol{\tau}$  for the moment):

$$x_{t,v} = \sum_k w_{t,k} z_{k,v} + \epsilon, \quad (2.51)$$

where  $\epsilon$  is a Gaussian error term like before. Thus, PPCA is similar to the PACA model, but with no prior distribution over  $\boldsymbol{\mu}$  and a Normal distribution over  $Z$  instead of a Gamma distribution.

Unlike PACA, PPCA violates two out of the three assumptions we have laid down for the model. Because  $\mathbf{W}$  has no prior distribution,  $\mathbf{W}$  is completely unconstrained and no regularization of the neural topics will occur. Furthermore, because the  $z$ 's are normally distributed, the model will learn more complex distributions over neural topics that will include many topics contributing negatively to the observed brain activity. Because of this, we expect that the reduced data  $\mathbf{W}$  and  $Z$  found by PCA on fMRI datasets will not capture underlying mental states as well as PACA, and the neural topics  $\mathbf{W}$  found by PCA will be difficult to interpret.

## 2.5.2 Non-negative Matrix Factorization (NMF)

Non-negative Matrix Factorization (NMF) (Lee & Seung, 1999) is a recently developed method of dimensionality reduction that solves the following problem, without a proba-

bilistic model: given a non-negative matrix  $V$ , find

$$V \approx WH, \quad (2.52)$$

where  $W$  and  $H$  are also non-negative matrices. Several equivalent iterative algorithms for solving this problem have been proposed (Lee & Seung, 2001). The fascinating property of NMF is that, when applied to a library of face images, the resulting reduced components appear to be easily interpretable “parts” of faces (figure 3.3), unlike the results of PCA, whose “eigenfaces” are often completely uninterpretable (figure 3.2).

Although NMF does not explicitly define a underlying probability model, Lee & Seung (2001) showed an iterative algorithm for solving NMF that solves the minimization problem,

$$\{W^*, H^*\} = \underset{W, H}{\operatorname{argmin}} \| \mathbf{X} - WH \|_2^2, \quad (2.53)$$

subject to the constraints,

$$W, H \geq 0.$$

Equation 2.53 is solving the same matrix factorization problem as PCA and NMF, but this time with the positive constraint on both matrix factors. Thus, when applied to neural data, NMF results in following factorization of a voxel’s activity  $x_{t,v}$ :

$$x_{t,v} = \sum_k w_{t,k} h_{t,k}, \quad x_{t,v} > 0, w_{t,k} > 0, h_{t,k} > 0 \quad (2.54)$$

Although the non-negative constraint on  $H$  satisfies the non negative constraint on  $Z$ , the non-negative constraint on  $\mathbf{X}$  and  $W$  clearly violates the neurological assumption that there should be inhibitory effects on brain activity evident in the data. Thus, it will be difficult for NMF to capture patterns of brain activity associated with cognitive states if inhibition is an important component of that activity. Second, the standard formulation of NMF does not include any regularization; we would therefore expect NMF to produce reduced datasets

that are “distracted” by noisy voxels and that fail to facilitate performance in the ‘mind-reading’ task.

## 2.6 Implementation

### 2.6.1 PACAfit and PACApredict

The basic model was implemented in a C program `PACAfit` on a UNIX system, using the multivariate minimization routines of the GNU Scientific Library (GSL) (Galassi et al., 2006). The GSL is a free numerical library licensed under the GNU General Public License (GPL). `PACAfit` uses the Fletcher-Reeves conjugate gradient algorithm (Fletcher, 1987) to minimize the negative log posterior (equation 2.26), using the parameterization of  $\mathbf{Z}$  given in equation 2.21 to convert the constrained optimization into an unconstrained optimization problem. The source code for `PACAfit` can be downloaded freely online at <http://www.princeton.edu/~djweiss/PACAfit/>.

A separate version of the program, `PACApredict`, was also created, for use in the reconstruction error cross validation experiments of section 3.3.2. `PACApredict` is identical to `PACAfit`, except that the gradient of the neural topics  $\mu$  is fixed to zero. Thus, `PACApredict` will fit brain states on a “test” set to neural topics found using `PACAfit` on a training set.

### 2.6.2 PCA and NMF

PCA was implemented in Matlab according to the method outlined by Jolliffe (2002). In this method, the eigenvectors of the sample covariance matrix  $\Sigma$  are computed and sorted by eigenvalue. To reduce the data to  $K$  dimensions, the top  $K$  eigenvectors are used to create a new basis,  $\Lambda$ . The reduced data is calculated by taking the linear projection of  $\mathbf{X}$  into the new basis defined by  $\Lambda$ .

To run NMF, we used the fast, Matlab-based implementation of NMF created by Lin (2007). This is a new algorithm that uses projected gradient methods to find solutions faster than the popular multiplicative update rules originally proposed by Lee & Seung (2001). When NMF was used in the cross validated reconstruction error experiments, the Matlab code was manually modified to remove the  $H$  update step, so that one of the non negative matrix factors could be fixed. The Matlab code for the projected gradients algorithm was downloaded from “<http://www.csie.ntu.edu.tw/~cjlin/nmf/>.”

### 2.6.3 Running Experiments

Datasets for the model were stored and processed in Matlab (Mathworks, Inc), using the Princeton Multi-Voxel Pattern Analysis (MVPA) Toolbox (Dretter et al., 2006). The MVPA Toolbox is a suite of Matlab scripts designed to facilitate rigorous pattern-recognition based analysis of fMRI datasets. Among other things, the MVPA Toolbox provides functions for Statistical Parametric Mapping (SPM), cross validated machine learning experiments, Z-scoring and detrending of neural data, and other pre- and post-processing techniques, as well as providing data structures to keep track of a large fMRI dataset.

For this thesis, a number of additional routines were added to the MVPA toolbox. The MVPA toolbox was modified to allow for the use of the external logistic regression package as a classifier, application of Principal Component Analysis (PCA) and Non-negative Matrix Factorization (NMF) to neural data, and a host of other minor utilities for managing large-scale experiments across clusters and reading and writing data to and from the PACA programs. (`PACAFit` was originally implemented in Matlab, but that proved to take too long to converge on even small datasets.) Altogether, the original Matlab code totaled approximately  $\sim 6000$  lines of code, while the C source code for `PACAFit` totaled  $\sim 1500$  lines of code.

## 2.6.4 Initialization & Convergence

In all of the experiments described in this thesis, the same initialization procedure was used. To initialize the neural topics, ten samples  $\{\mathbf{X}_{t_0}, \dots, \mathbf{X}_{t_{10}}\}$  were chosen from dataset at random with replacement. We then averaged these together and smoothed by a factor of four to initialize the first topic. The random sampling and smoothing procedure was then repeated for each topic  $\mu_k$ . To initialize the brain states, each  $z_{k,t}$  was set to  $1/K$ . The initialization procedure was decided upon after a number of brief experiments investigating the time until convergence for a set of different random initialization experiments.

To assess convergence, the ratio  $\rho$  of the log posterior over the average of the log posterior of the last 10 rounds was used as a criterion:

$$\rho = \frac{\mathcal{L}(\hat{\theta}_t; \mathbf{X})}{\frac{1}{10} \sum_{i=1}^{10} \mathcal{L}(\hat{\theta}_{t-i}; \mathbf{X})} \quad (2.55)$$

As a general rule, we considered the algorithm to have converged if the ratio  $\rho < 5 \times 10^{-6}$ . The ratio to average of the last 10 iterations was used instead of the single last iteration because of the brief “lulls” observed in the progress of the iteration, that would produce an artificially low estimate of the change in log posterior each round (figure 3.1).

Generally, the algorithm converged in less than 700 iterations, across all datasets. The time required to fit PACA on a 2 GHz PowerPC G5 cluster computer ranged from approximately 10-40 minutes, depending on the value of  $K$  and the dimensionality of the dataset.

# Chapter 3

## Results

In this chapter, we apply the methods developed in chapter 2 to a number of different problems. Because matrix decompositions of natural images allow easy visualization of the components, we begin by applying PACA to a publicly available library of faces in section 3.1. The face libraries allow us to gain an intuition of how PACA decomposes images in comparison to PCA and NMF. We then apply PACA to two representative, real-world fMRI datasets: a block-design image viewing task used in Haxby, et al. (2001) (section 3.2.1), and a word encoding task in currently ongoing psychological research (Robison et al., 2007).

### 3.1 CBCL Face Library

Before we apply PACA to the mind-reading problem, we consider dimensionality reduction of face databases. The relation to the fMRI dimensionality reduction is straightforward: instead of a collection of 3D MR images consisting of voxels, we have a collection of 2D photographic images consisting of pixels. Because none of the methods in this paper assign meaning to neighboring spatial relationships, we can simply consider the  $v$ 'th voxel to be

the  $v$ 'th pixel, and write

$$x_{t,v} = \text{brightness of pixel } v \text{ in image } t.$$

Like the neural topics, the pixel topics  $\mu_k$  found by the algorithm will be images themselves. For simplicity, we refer to the pixel topics as *basis images*.

We applied both variants of PACA, PCA, and NMF to face images from the CBCL face library (Massachusetts Institute of Technology), the same freely available database used in the original presentation of NMF (Lee & Seung, 1999). This library consists of 2901 19x19 grayscale images of faces. The images have been adjusted so that the location of the head is scale and position invariant, but there are multiple images of each subject, taken with different facial expressions. We used all images in all of our analyses. To ensure that the data was consistent with our modeling assumptions, the images were z-scored along each pixel dimension before analysis. The reconstructed images were re-scaled and un-centered before being displayed. Before running NMF, the z-scored data was shifted so that the minimum value was exactly equal to zero.

### 3.1.1 Convergence

We assessed the convergence of the PACA algorithm by examining the value of the log posterior,  $\mathcal{L}$ , as a function of the iterative steps taken by the conjugate gradient descent procedure. These traces are displayed for two values of  $K$  in figure 3.1. It is clear from the figure that convergence on this dataset with our initialization scheme was very stable; two restarts were used in all subsequent analysis unless specified otherwise.

### 3.1.2 “Additive Face Components”

Although PACA, PCA, and NMF are all able to qualitatively reproduce the faces in the database with a relatively small number of components, there is a marked difference be-

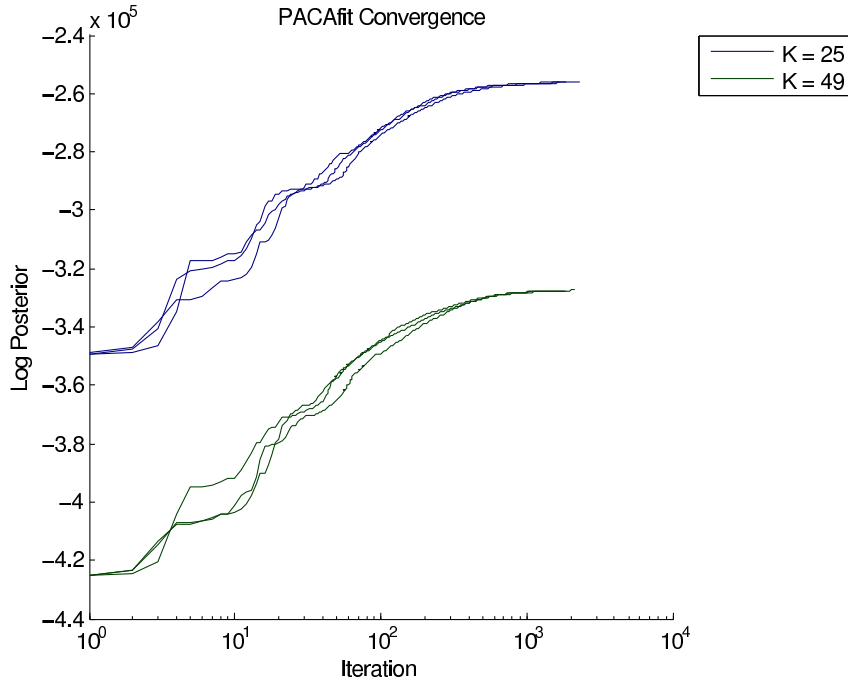


Figure 3.1: PACA Convergence on the CBCL Face Library dataset. Each line represents one restart. Hyperparameters:  $\lambda = 0.01$ ,  $\gamma = 0.01$ .

tween the basis images found by the different algorithms. The basis images found by PACA, PCA, and NMF are displayed in figures 3.2, 3.3, and 3.4 for comparison. In figure 3.4, the hyperparameters for PACA are  $\lambda = 0.01$ ,  $\gamma = 0.01$ . As described in Lee & Seung (1999), the PCA basis images (or “eigenfaces”) bear vague resemblance to aspects of faces, such as shading, but are generally hard to interpret. This is apparent in figure 3.2a, as the PCA basis images become progressively “noisier” as the eigenvalue of principal component decreases (bottom right). In contrast, because of the non-negativity constraints in NMF, the NMF basis images are more natural “parts” of faces that are easy to interpret. As can be seen in figure 3.3a, the NMF basis images, though not entirely anatomical, do contain “pieces” that generally correspond to various areas of the face. In addition, the representation of images in the new basis found by NMF is much sparser than that found by PCA (figure 3.3b).

In comparison to both PCA and NMF, the basis images found by PACA appear to be somewhat of a compromise between individual and global features of the dataset. Like

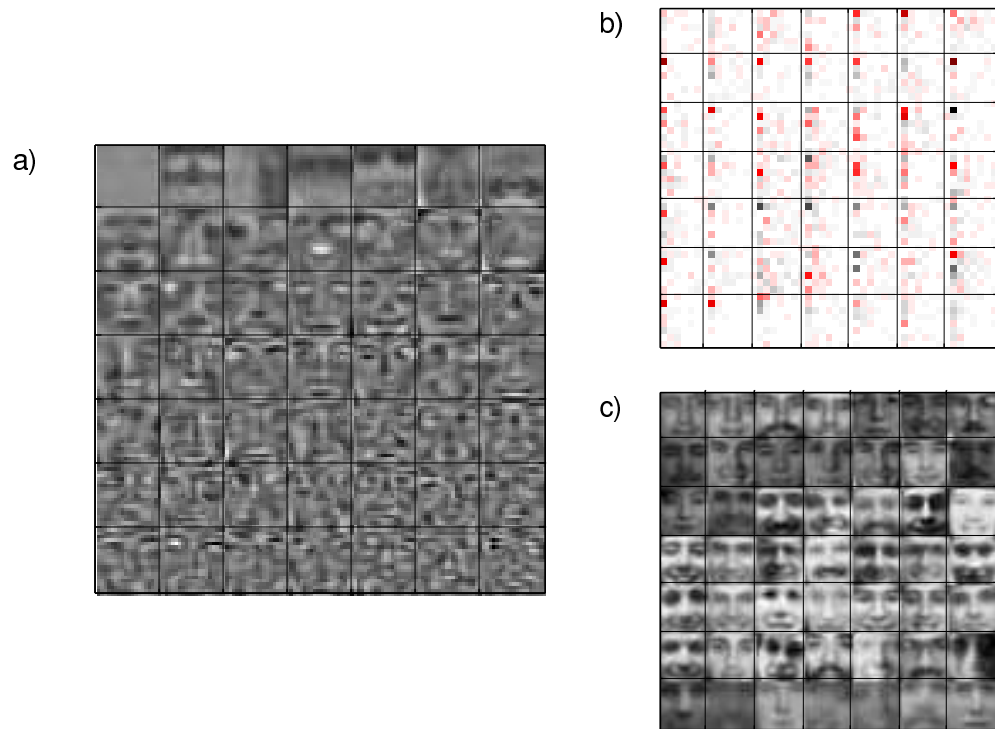


Figure 3.2: PCA based decomposition of the CBCL face library. **a**, The 49 eigenfaces found by the decomposition. Negative values are black, neutral gray, and positive white. **b**, The coefficients used by the decomposition to reconstruct the images in the library. Negative values are red, neutral white, and positive black. Each pixel in each image corresponds to the basis in **a** at the corresponding location. **c**, The reconstructed faces from each cell in part **b**.

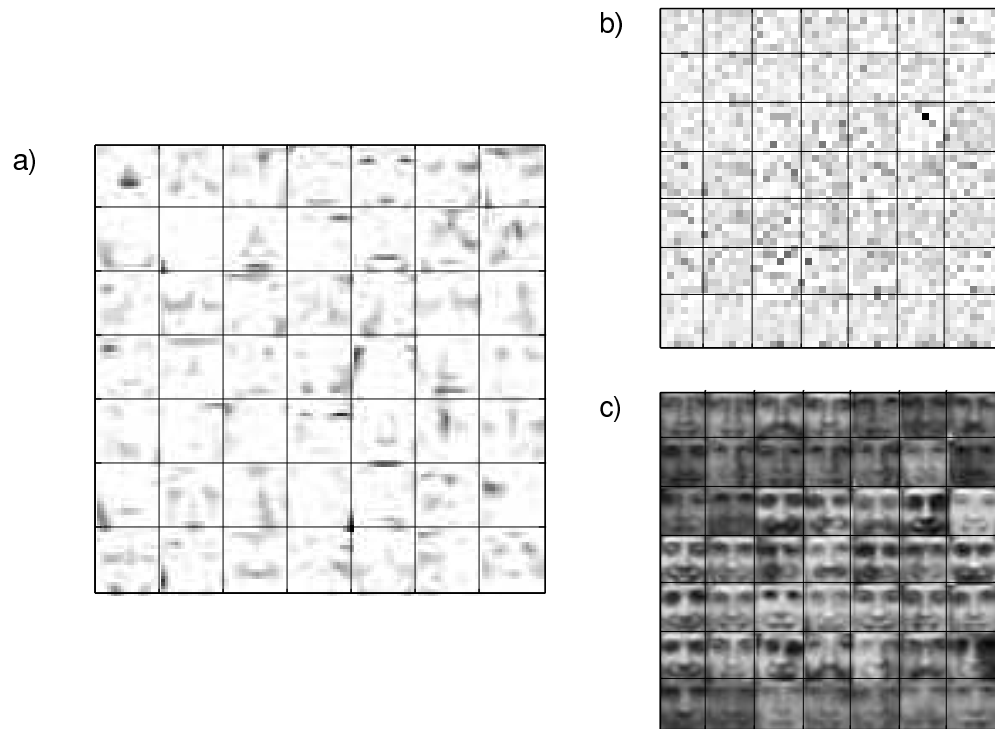


Figure 3.3: NMF based decomposition of the CBCL face library. **a**, The 49 face “parts” found by the decomposition. Positive values are black; small values are white. **b**, The coefficients used by the decomposition to reconstruct the images in the library. Positive values are black; small values are white. Each pixel in each image corresponds to the basis in *a* at the corresponding location. **c**, The reconstructed faces from each cell in part **b**.

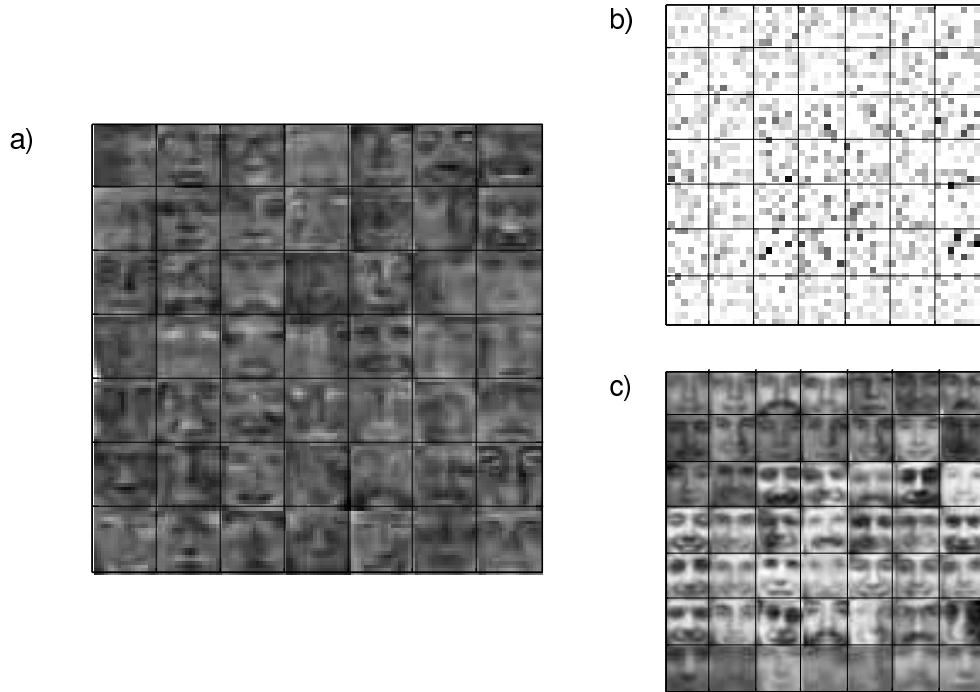


Figure 3.4: PACA based decomposition of the CBCL face library ( $\lambda = 0.01, \gamma = 0.01$ ). **a**, The 49 additive face components found by the decomposition. Negative values are black, neutral gray, and positive white. **b**, The coefficients used by the decomposition to reconstruct the images in the library. Negative values are red, neutral white, and positive black. Each pixel in each image corresponds to the basis in **a** at the corresponding location. **c**, The reconstructed faces from each cell in part **b**.

NMF, the representation of images in the new basis space is relatively sparse compared to PCA (figure 3.4b), and such sparse encoding implies that the bases themselves must encode more information about aspects of individual faces. However, unlike NMF, the representations are not totally sparse when the hyperparameter  $a > 1$ , since the barrier function in the log likelihood prevents any of the component activations  $z_{k,t}$  from being exactly zero. Thus, if we examine the basis images found by PACA, or the “additive face components,” we see that some of the basis images qualitatively appear to represent global aspects of the face database, such as lighting highlights and general facial expression, while others appear to represent components of individual subjects, such as glasses (figure 3.4a).

As we increase the regularization parameters  $\gamma$  and  $\lambda$ , the priors on  $\mu$  and  $\mathbf{Z}$  serve to increase the smoothness of the face components and to increase the relative sparsity of the

activations, so the trade-off between sparse and dense encoding of faces becomes more apparent. Figure 3.5 shows the decomposition of faces with  $\lambda = 5$  and  $\gamma = 5$ . Here, the difference between global features of faces and individual features of faces is very clear: in the set of face components (figure 3.5a), there are a few very large amplitude components that appear to correspond to global features of the data, and many more very small amplitude faces that appear to correspond to individual features of sets of subjects. If we look closely at the component activations of a single face, we see that the “global” features have small values of  $z_{k,t}$ , while a sparse set of “individual” features have large values, with the remaining “individual” features having small values. Thus, because of the large amplitude of the “global” features in  $\mu$ , a small value in  $\mathbf{Z}$  is still sufficient to recreate the global aspects of a particular image. And because of the small amplitude of the “individual” features in  $\mu$ , only the large-valued “individual” features provide a significant contribution to the reconstructed image. Thus, shape of the prior distribution over  $\mathbf{Z}$  when  $a > 1$  causes PACA to find a representation of the data using both large-amplitude, “global” trends in the data and small-amplitude, “local” features of individual observations. In contrast, if we use the same high values of  $\gamma$  and  $\lambda$  but use the more strict prior over  $\mathbf{Z}$ , when  $a \leq 1$ , the resulting component activations are so sparse that only the “global”, average features of the data are used, and the reconstruction of the original faces is very poor (data not shown).

## 3.2 Human fMRI Datasets

After investigating the qualitative properties of the components discovered by the PACA model on the face datasets, we applied the PACA model to real-world fMRI data used from two “mind-reading” experiments. Although the goals of the experiments are different, they both involve the classification paradigm described in section 1.1. From the first fMRI dataset, we also created a set of synthetic data that we use to investigate the specific

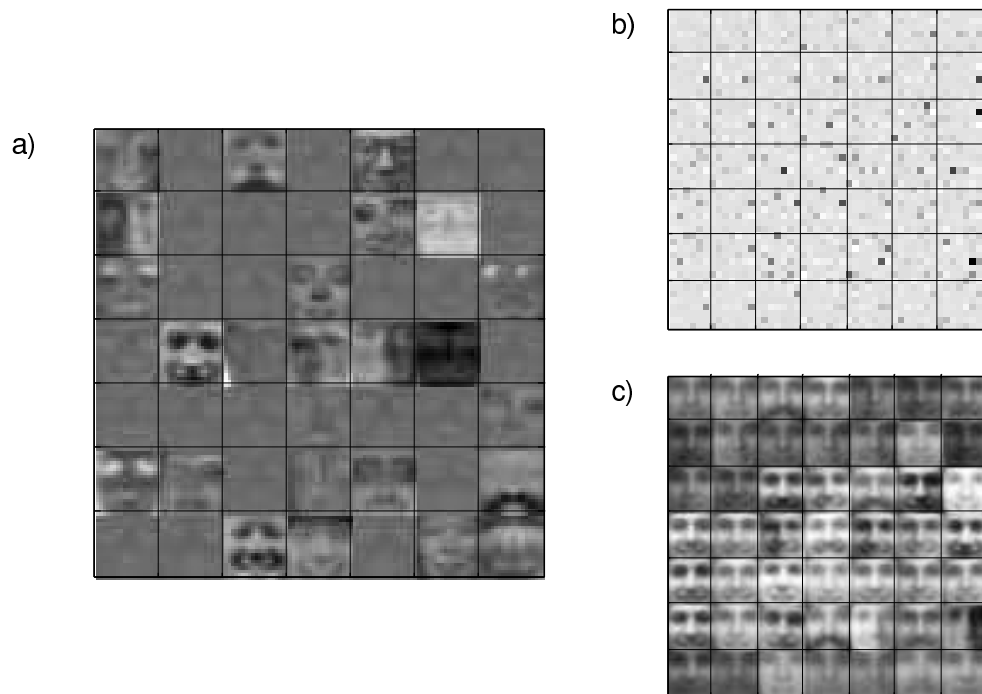


Figure 3.5: PACA based decomposition of the CBCL face library, with high regularization ( $\lambda = 5, \gamma = 5$ ). The high regularization forces separation of “global” and “individual” details from the images. **a**, The additive face components found by the decomposition. Negative values are black, neutral gray, and positive white. **b**, The reconstruction coefficients found by the decomposition; small values are white, positive values are black. **c**, The reconstructed faces from part **b**.

conditions in which PACA results in improved analysis over other algorithms. We now briefly summarize the important details and qualities of each of these datasets.

### **3.2.1 Face and Object Representations (Haxby et al., 2001)**

The first fMRI dataset we used in our analysis was from the experimental data used in Haxby et al. (2001). In this experiment, six subjects viewed photographs from eight categories (faces, cats, houses, chairs, scissors, shoes, and bottles, and control, non-sense images) while in an fMRI scanner. The original experimental analysis showed that the different classes of objects elicited different patterns of large and small responses across a wide area of cortex, supporting a model of cortex in which objects are represented by a topographically organized distribution of attributes (Haxby et al., 2001). This study was one of the first “mind-reading” experiments, as the image category was predicted significantly on one half of the data using correlation-based similarity measures computed from the image categories in the other half.

For our analysis, we obtained the data from one subject in the experiment. The experiment consisted of 10 trials (or “runs”), each containing a single 24s stimulus block ( 9-10 recorded datapoints) for each category (Haxby et al., 2001). Because of the involvement of ventral temporal cortex in high-level object recognition (Haxby et al., 2001), we restricted all analysis to this area. To determine which voxels to include in the anatomical region, we used a brain atlas in Talairach space to select voxels in the region consisting of the lingual, parahippocampal, fusiform, and inferior temporal gyri, consistent with the definition in Haxby et al. (2001). After stripping out unwanted voxels and applying the preprocessing and averaging steps of our analysis (section 3.3), the single subject’s dataset consisted of 80 averaged timepoints and 3379 voxels, with 10 timepoints for each category. The entire brain area in this dataset consisted of 43,193 voxels.

### 3.2.2 Retrieval Orientation State Memory (Robison et al., 2007)

The second set of fMRI datasets we used in our analysis was from a currently ongoing study continuing research in classifying “retrieval orientation” state. These experiments involve guiding subjects to memorize words while using specific mnemonic devices, and then testing subjects’ memory of the words using one of the mnemonics as a cue. The “retrieval orientation” is the cue given to the subject to recall the word, which has been shown to bias the accuracy of the recall process (Robison et al., 2007). In the Robison et al. (2007) experiment, subjects are asked to study noun words while performing one of three mnemonic tasks: *artist*, *function*, and *read*. In the artist task, subjects were asked to rate how difficult it would be to draw the object; in the function task, subjects were asked to generate lists of potential uses for the object; and in the read task, subjects were asked to read the word backwards silently (Robison et al., 2007). In the second phase of the experiment, the goal was to test subjects’ retrieval orientation state memory (ROSM), by showing them a combination of new and old words and asking them to indicate whether or not a particular mnemonic was used on a given word (Robison et al., 2007).

For our analysis, we obtained the data from six of the subjects in the experiment during the study phase. Each dataset consisted of six runs in which the subjects were presented with 9 words of each task in mini-blocks of 3 words. Each word was presented for 4 seconds (two fMRI timepoints). Because the frontal lobe is thought to be responsible for top-down mechanisms mediating memory retrieval, analysis was restricted to voxels within the frontal lobe of each subject. Frontal lobe anatomical masks were provided by Robison et al.. After stripping out unwanted voxels and applying the preprocessing and averaging steps of our analysis (section 3.3), each subject’s dataset consisted of 57 averaged timepoints (19 for each task) and between roughly 8,000 and 10,000 voxels, depending on the subject. The entire brain in these subjects ranged from 43,000 to 58,000 voxels. A summary of the dimensionality of each subject is given in table 3.1.

Experiment	Subject	Total $V$	Masked $V$	$T$
Haxby et al. (2001)	haxby1	43,193	3,379	80
Robison et al. (2007)	rosm05 <sup>5</sup>	48,475	8,757	57
Robison et al. (2007)	rosm07	53,904	10,581	57
Robison et al. (2007)	rosm08	43,504	7,905	57
Robison et al. (2007)	rosm13 <sup>56</sup>	58,473	10,667	57
Robison et al. (2007)	rosm14	41,920	7,926	57
Robison et al. (2007)	rosm15 <sup>5</sup>	57,154	10,575	57

Table 3.1: Summary of the different experimental fMRI datasets used in the analysis.

### 3.2.3 Synthetic Datasets

To assess whether or not the PACA model could extract more information from a set of highly informative voxels than the uni-variate SPM approaches, we created an “ideal” synthetic dataset from the Haxby et al. (2001) dataset. This dataset consists of 577 voxels determined to be “category selective” using the criteria discussed in Haxby et al. (2001); this dataset is considered “synthetic” because these voxels were selected using the entire dataset. Thus, using this dataset for cross validation is “cheating,” as the supervised voxel selection process to build the data cheated by looking at both the training and test sets of the data. However, the dataset is still useful to a certain extent, because this dataset represents an “ideal” scenario, in which the observed data consists of a small set of highly informative voxels.

## 3.3 Assessing Generalization Using Cross Validation

To assess the utility of the dimensionality reduction method, we recreated the entire analysis procedure of a typical ‘mind-reading’ experiment (figure 3.6) for each of the fMRI datasets. First, to eliminate the intrinsic drift in the fMRI signal from run to run, each voxel was z-scored (subtract the mean, divide by standard deviation) for each recorded run sep-

---

<sup>5</sup>Subject was indicated to have “good” generalization from the study phase of the ROSM experiment to the test phase (Robison, personal communication).

<sup>6</sup>Subject was the author of this thesis.

arately (figure 3.6b). Second, all consecutive stimulus presentations belonging to a single cognitive category (or “miniblocks”) were averaged together, and all “rest” timepoints not involved in a stimulus presentation were removed (figure 3.6c). Because the presentation of mini-blocks was randomized in each experiment, the averaging in the second step served to remove all time-series dependencies from the data. (This was necessary to satisfy the assumptions of exchangeability described in chapter 2).

After preprocessing, cross-validation was used to assess two types of generalization performance metrics on each dataset: reconstruction error (figure 3.6d) and classifier prediction error (figure 3.6e). In the reconstruction error paradigm, we separated the data into two halves based on odd and even runs. We then ran `PACAFit` on one half of the data (the “training set”), and then ran `PACAPredict` on the other half (the “test” set) with  $\mu$  fixed to the values estimated by `PACAFit` on the first half. We could then compute the root mean squared reconstruction error (RMSE),

$$\text{RMSE}(\mathbf{X}, \hat{\mathbf{X}}) = \sqrt{\frac{1}{VT} \sum_{t,v} (x_{t,v} - \hat{x}_{t,v})^2}, \quad (3.1)$$

where  $\mathbf{X}$  is the observed voxel activity on the test set and

$$\hat{\mathbf{X}} = \mathbf{Z}_{\text{test}}^T \boldsymbol{\mu}_{\text{train}}, \quad (3.2)$$

is the predicted voxel activations on the test set using the neural topics fit to the training set. The RMSE is thus a measure of the ability of the PACA model to find neural topics that explain more than the dataset used to generate those topics, regardless of the context of the experiment.

To compare the RMSE of PACA against similar values for PCA and NMF, slightly different procedures were necessary. Since PCA can only reduce the dimensionality of the data to  $T$  components at a maximum, training PCA on only half of the data would greatly limit the potential number of principal components, so the  $N - 1$  cross validation

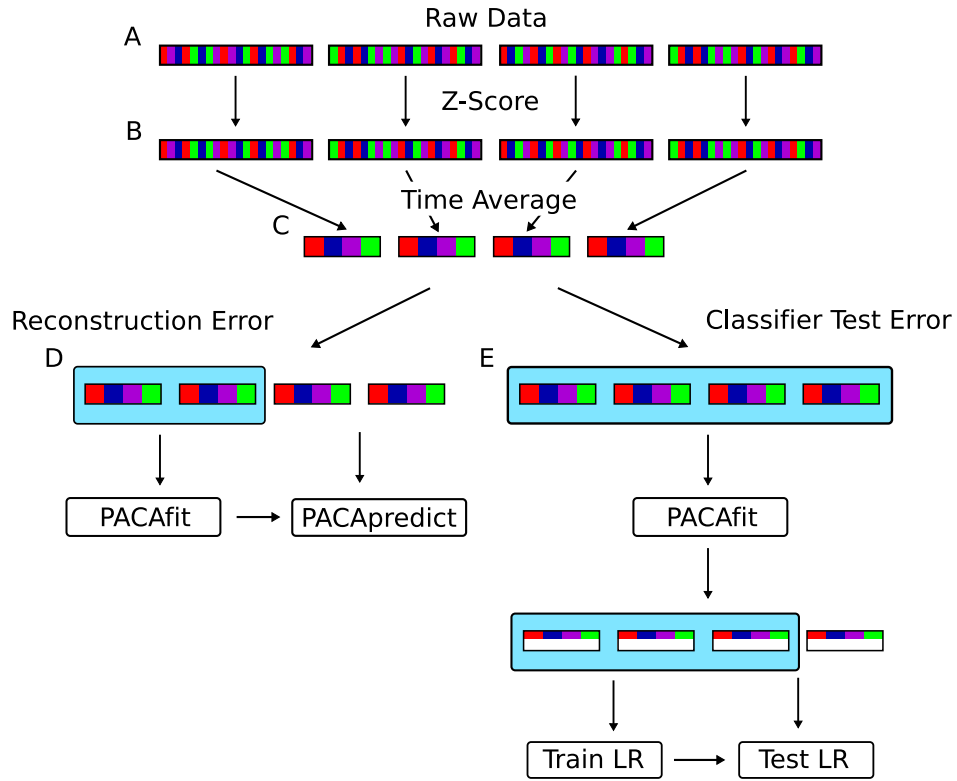


Figure 3.6: Outline of the data processing pipeline used in analysis of fMRI experimental data analysis in this thesis. **a**, The raw fMRI data is recorded in experimental blocks called “runs”, in which each experimental condition is presented in random order a fixed number of times. Each presentation lasts for several timepoints, and the conditions are represented in the graph by color. Timepoints not involved in an experimental condition are excluded from the analysis. **b**, Due to drift in the fMRI BOLD signal from run to run, each voxel is z-scored separated for each run. **c**, To remove dependence between timepoints, all the timepoints within a given block of stimulus presentation are averaged together. **d**, To test the reconstruction error of PACA, the algorithm is first run on half of the data (blue highlight). Using the components  $\mu$  found by the algorithm in the first step, PACA finds the reduced matrix  $Z$  for the held-out set. The reconstructed error is then calculated from the difference  $Z^T \mu - X_{test}$ . This whole process is then repeated with the roles of the training set and test set switched (not shown). **e**, To test PACA as a dimensionality reduction method in “mind-reading” classification experiments, a  $N - 1$  cross-validation classification experiment is run after transforming the entire dataset into the PACA reduced space. For each run  $n$  of  $N$  runs, the  $n$ 'th run is held out and used as a test set, and a state-of-the-art logistic regression (LR) classifier is trained to predict the experimental conditions from the averaged brain images in the training set. Only the first iteration is shown.

method was used (see below). To obtain the RMSE score for PCA, the voxel predictions  $\hat{\mathbf{X}}$  were obtained through the linear projection of the test set onto the principal components calculated from the training set. To obtain RMSE scores for NMF, the entire dataset (both training and test) was shifted to become positive. Furthermore, the NMF algorithm in (Lin, 2007) was modified to remove the update step of one of the factors, so that the factorization could be generalized from the training set to the test set.

To assess the usefulness of the reduced data as a means of feature selection in a “mind-reading” experiment, we also measured the classifier prediction error (figure 3.6e). In the classifier prediction error paradigm, we first ran PACA (or one of the competing algorithms) on the entire subject’s dataset. We then trained a state-of-the-art logistic regression classifier (Paul Komarek, 2005) to predict the cognitive labels from the reduced data using an  $N$ -fold, leave-out-one cross validation procedure, where  $N$  is the number of trials or runs of the experiment. The procedure for leave-out-one cross validation is straightforward: for each of  $N$  runs, choose one run to use as the test set, and train the classifier on all of the others. Next, measure the generalization error of the classifier on the held-out run. The final estimate of the error of the classifier’s predictions is the average of the test error across all  $N$  rounds of the procedure. Thus, for the Haxby et al. (2001) dataset, we used 10 rounds of cross-validation; for the ROSM datasets, we used 6 rounds of cross-validation.

Finally, for a comparison in general of the unsupervised dimensionality reduction methods to existing supervised methods for feature selection, we ran a traditional SPM-based cross-validation experiment as well for each dataset. Because SPM is a supervised method, it must be performed separately on the training set for each round of cross validation. In our analysis, we performed SPM by running a uni-variate ANOVA to determine which voxels varied significantly across cognitive labels in the training set. We then selected the  $V$  voxels with the smallest ANOVA  $p$ -values, where  $V$  was the desired reduced number of voxels. We then use this much smaller subset of voxels as the input to the classifier. Thus, a different subset of  $V$  voxels is used on each round of the cross-validation classification

experiment.

### 3.3.1 Avoiding Sparse Brain States

As described in section 2.4.3, if the PACA hyperparameter  $a$  is less than or equal to one, then the Gamma prior over  $\mathbf{Z}$  will cause the estimates of  $\mathbf{Z}$  to be very sparse. Since there is evidence for distributed and overlapping representations of objects in cortical areas (Haxby et al., 2001), we expect that forcing the model to find sparse brain states will result in a much poorer reduced dimensionality representation of the data. Qualitatively, running PACA on the face datasets with the sparse Gamma prior reduced the recognizability of the reconstructed faces (section 3.1). To quantitatively assess the effect of a sparse prior over brain states, we examined the generalization performance of the PACA reduced data in both the reconstruction error and classifier prediction error paradigms (figure 3.7) over a large set of parameters. In both cases, the generalization performance of PACA when  $a = 1$  was greatly reduced to the case where  $a > 1$ .

As a consequence, we restricted the rest of our analysis to the  $a > 1$  case. Furthermore, based on the initial experiment and other exploratory analysis (data not shown), we found that the set of hyperparameters covering all combinations of  $\gamma, \lambda = [0.1, 0.01]$  showed the performance of the algorithm with relatively low, medium, and high amounts of regularization. All subsequent analysis was performed using these sets of parameters unless otherwise noted.

### 3.3.2 Reconstruction Error

The results of analysis using the reconstruction error paradigm (figure 3.6d) are shown for the Haxby et al. (2001) in figure 3.8 and for three subjects from the ROSM dataset in 3.9. We restricted our analysis to three ROSM subjects due to the large amounts of time required to run the reconstruction error paradigm. The number of the ROSM subject is indicated in the upper left corner of each subplot in figure 3.9. In each experiment, we compare the

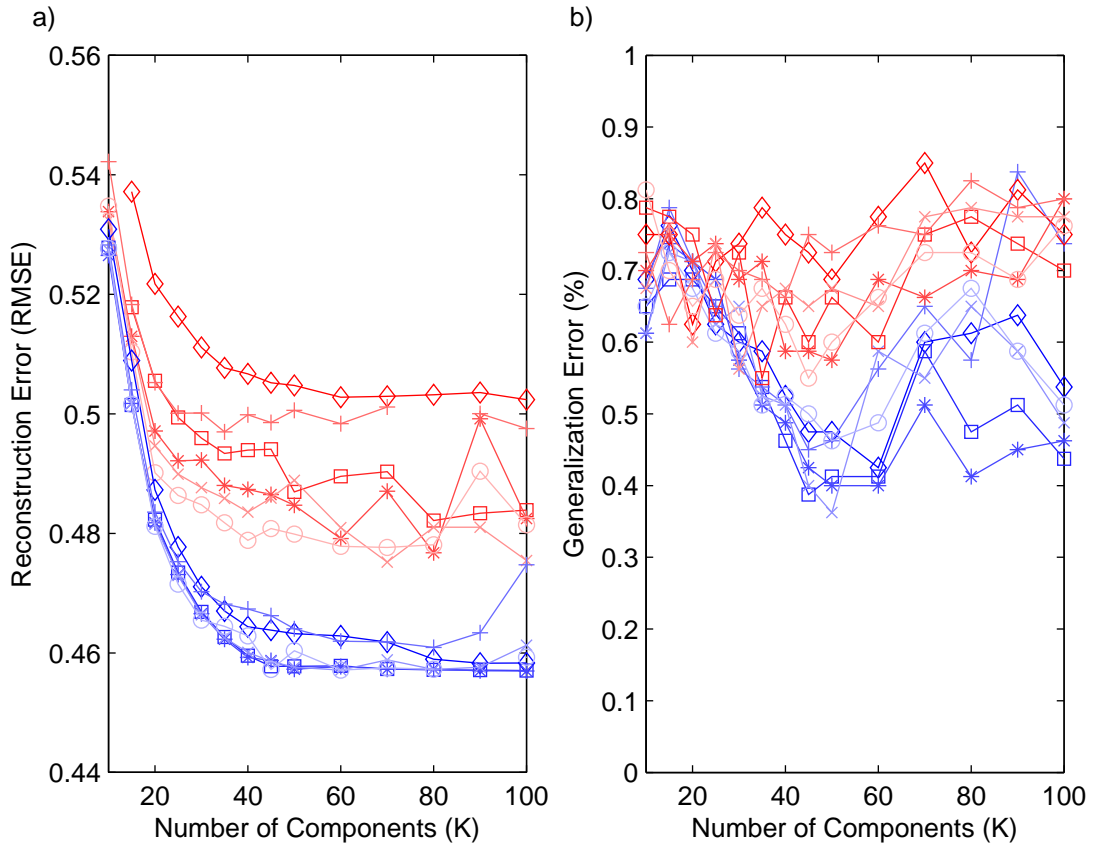


Figure 3.7: Comparison of both cross validated reconstruction error, **a**, and classifier prediction error, **b**, with smooth (blue) vs. sparse (red) priors over  $\mathbf{Z}$ . PACA was run according to the protocol in figure 3.6 with six sets of hyperparameters corresponding to all combinations of  $\lambda = [0.01, 0.1, 1]$  and  $\gamma = [0.01, 0.1]$ . The shade of the lines indicates the relative amounts of regularization (light is low regularization, dark is high regularization). Blue datapoints were obtained from PACA using the reparameterization with  $a > 1$ , while red datapoints were obtained from PACA using the reparameterization with  $a = 1$ . All data is shown from the single subject from the Haxby et al. (2001) dataset.

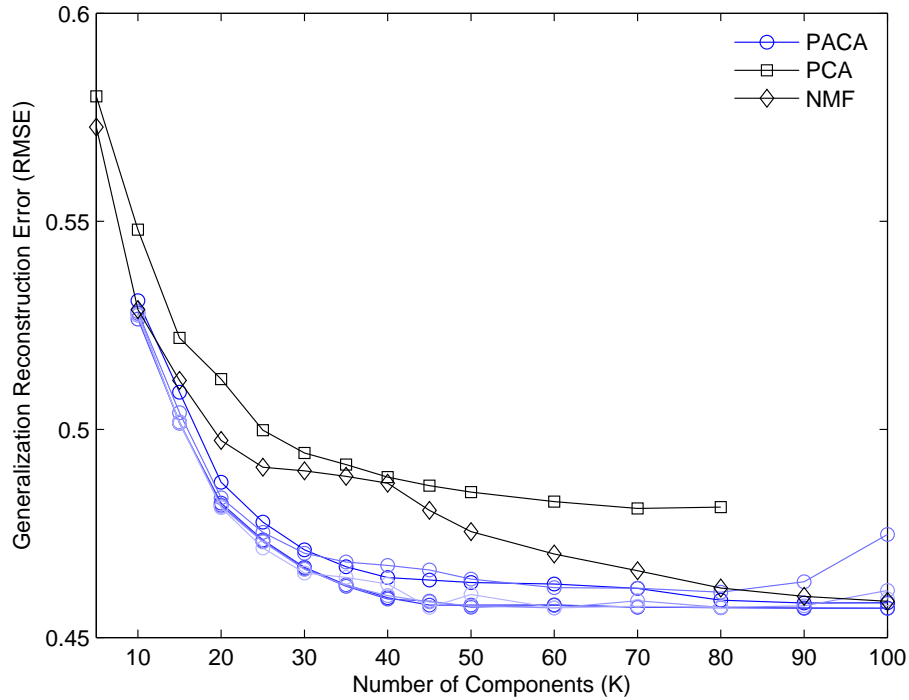


Figure 3.8: Cross validated reconstruction error on the Haxby et al. (2001) dataset. PACA (circles), PCA (squares), and NMF (diamonds) were run with  $K$  ranging from 5 to 100. PACA was run with four settings of  $(\lambda, \gamma)$ : (0.01, 0.01) (lightest blue), (0.01, 0.1) (light blue), (0.1, 0.01) (blue), (0.1, 0.1) (dark blue). The measured performance metric was RMSE (lower is better).

performance of PCA, PACA, and NMF with values of  $K$  ranging from 5 to 100. We note that in the case of  $K > 57$  in the ROSM data and  $K > 80$  in the Haxby et al. (2001) data, the matrix of topic patterns  $\mu$  is actually larger than the original dataset  $\mathbf{X}$ , even though the dimensions of  $\mathbf{Z}$  (the “reduced data”) are much smaller. In these cases, the term “dimensionality reduction” is technically a misnomer, since we cannot recreate  $\mathbf{X}$  without the matrix  $\mu$ . Nonetheless, the decompositions are still neuroscientifically interesting, since from the perspective of the classifier, only the dimensionality of the reduced data  $\mathbf{Z}$  is important.

Overall, the RMSE of NMF and PACA is lower than that of PCA in almost all cases, while NMF and PACA have similar reconstruction errors in the ROSM experiment and different performances in the Haxby et al. (2001) experiment. As can be seen in figures

3.8 and 3.9, PCA is in all but one case far from either PACA or NMF. Quantitatively, the mean of PCA over all subjects and values of  $K$  was  $.507 \pm 0.003$ , while the overall means of NMF and PACA were  $.489 \pm 0.004$  and  $.482 \pm 0.002$  respectively. The differences between average PACA RMSE and average NMF or PCA RMSE was highly significant difference in both cases (two sampled T-test<sup>7</sup>,  $p < 1 \times 10^{-5}$ ). However, although the RMSE of PACA and NMF differed significantly ( $p < 0.001$ ) across all parameter sets when considering the Haxby et al. (2001) dataset alone, there was no significant difference between NMF and PACA within the ROSM experiments alone ( $p = 0.66$ ). Rather, there was a massive difference in the *training* RMSE between PACA and NMF (data not shown): the mean training RMSE of NMF across all subjects and parameter sets was  $0.081 \pm 0.013$ , while the mean training RMSE of PACA was  $0.409 \pm 0.013$ , a highly significant difference ( $p < 1 \times 10^{-15}$ ).

However, the PCA results are somewhat unreliable, as the large disparity between PCA and the other algorithms apparent in ROSM subjects 5 and 15 (figure 3.9) is not apparent in the Haxby et al. (2001) data or in ROSM subject 13. If we exclude subjects ROSM 5 and 15 from our analysis, there is a significant difference between the RMSE of PCA and PACA across all parameter sets and subjects ( $p < 1 \times 10^{-7}$ ). Furthermore, if we exclude ROSM subjects 5 and 15, there is no longer a significant difference between the RMSE of NMF and PCA ( $p = 0.95$ ), and the RMSE of PCA is significantly lower than both NMF and PACA (PCA =  $0.508 \pm 0.004$ , NMF =  $0.507 \pm 0.004$ , PACA =  $0.482 \pm 0.003$ ,  $p < 1 \times 10^{-5}$ ). We can therefore conclude that, regardless of the strange variation in the PCA results, PACA has significantly lower reconstruction error than PCA in all cases.

### 3.3.3 Classifier Prediction Error

The results of analysis using the classifier prediction error paradigm (figure 3.6e) are shown for the Haxby et al. (2001) dataset in figure 3.10 and for six subjects from the ROSM dataset

---

<sup>7</sup>Null hypothesis: the mean RMSE values of two compared algorithms are the same.

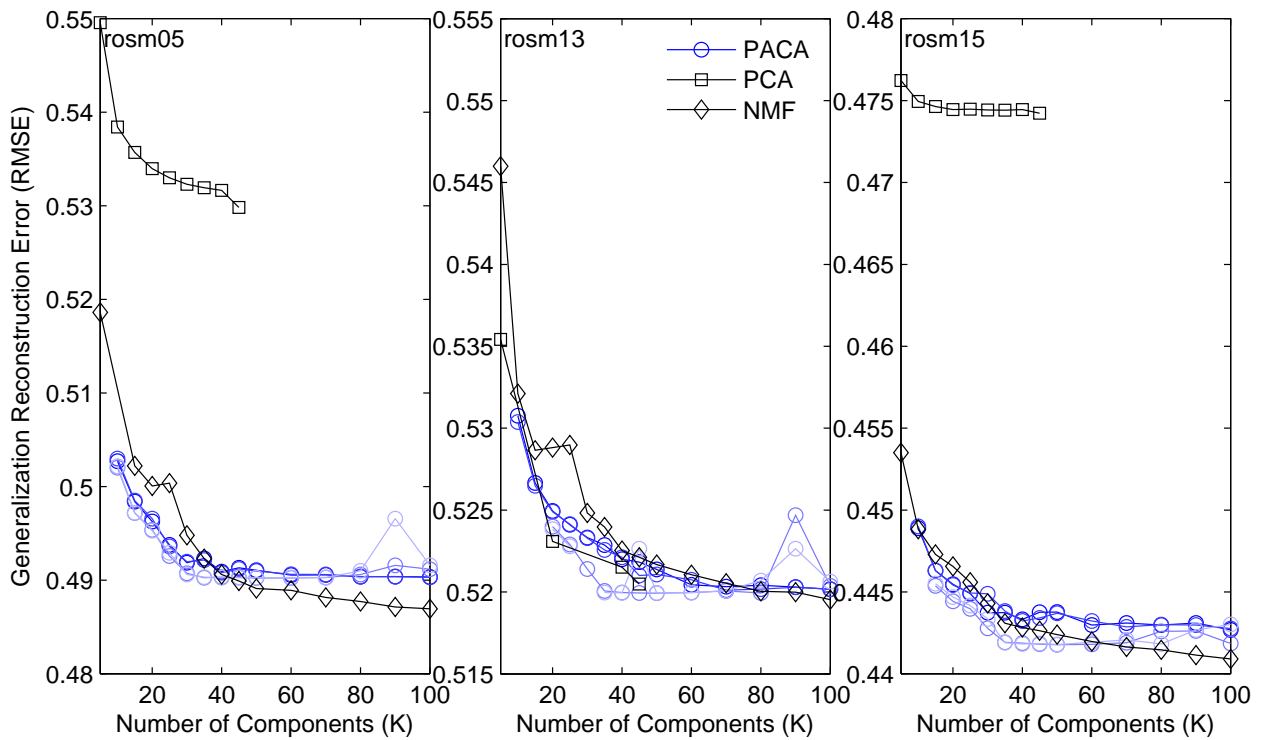


Figure 3.9: Cross validated reconstruction error on the ROSM dataset. PACA (circles), PCA (squares), and NMF (diamonds) were run on each subject with  $K$  ranging from 5 to 100. PACA was run with four settings of  $(\lambda, \gamma)$ : (0.01, 0.01) (lightest blue), (0.01, 0.1) (light blue), (0.1, 0.01) (blue), (0.1, 0.1) (dark blue). The measured performance metric was RMSE (lower is better). The identity of each subject is indicated in the upper left corner of the subplots.

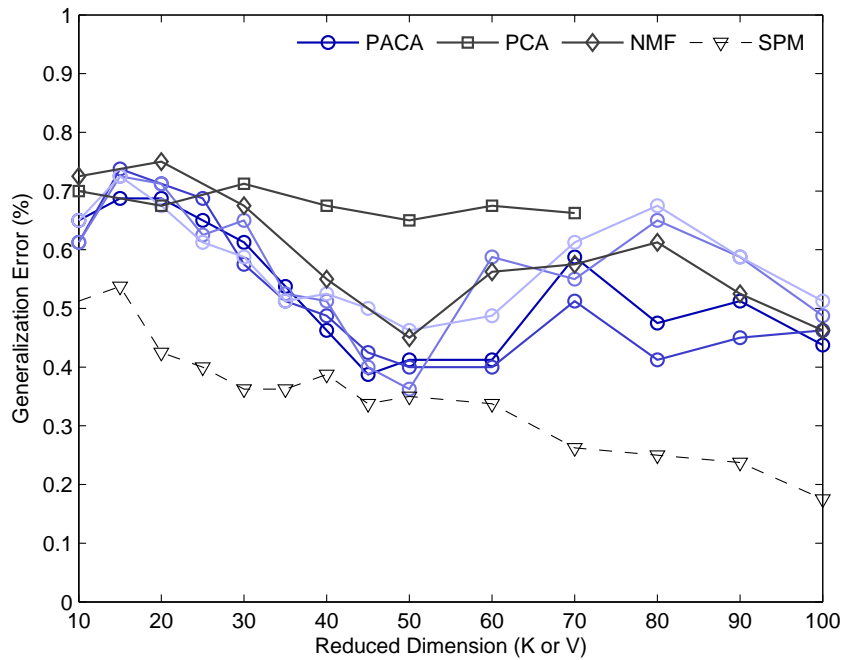


Figure 3.10: Cross validated classifier prediction error on the Haxby et al. (2001) dataset. PACA (circles), PCA (squares), NMF (diamonds), and SPM (dotted triangles) were each run with  $K$  ranging from 5 to 100. PACA was run with four settings of  $(\lambda, \gamma)$ : (0.01, 0.01) (lightest blue), (0.01, 0.1) (light blue), (0.1, 0.01) (blue), (0.1, 0.1) (dark blue). A logistic regression classifier was trained on the reduced data to predict cognitive labels for each round of cross validation. The measured performance metric is the average error rate of the classifier on the test set across all rounds.

in figure 3.11. For each dataset, we compared classification on reduced data obtained by SPM, PACA, PCA, and NMF, with  $K$  (or  $V$  in the case of SPM) ranging from 5 to 100. The same technical considerations about dimensionality reduction discussed in section 3.3.2 apply to this section also. We measured accuracy using the average percentage of incorrectly labeled examples on the test set, as discussed in section 3.3. In figure 3.11, the identity of the ROSM subject is indicated in the upper left corner of the subplot. The lightness of the PACA line indicates the relative regularization of the PACA model used to obtain datapoints in the line, ranging from light shading for the lowest ( $\lambda, \gamma = 0.01$ ) to dark shading for the highest ( $\lambda, \gamma = 0.1$ ).

Although none of the supervised algorithms consistently outperformed SPM, PACA for dimensionality reduction performed significantly better than both PCA and NMF in almost all cases. Qualitatively, we see in figures 3.10 and 3.11 that PACA with relatively high regularization (dark blue lines) has lower generalization error than both PCA and NMF. Performance tends to peak in most subjects in the range  $K = [30, 50]$ . Quantitatively, the average generalization error across all subjects and values of  $K$  of PACA with high regularization over topics ( $\lambda = 0.1, \gamma = 0.01$ ) was  $36.1\% \pm 1.72$ , while average generalization error of PCA was  $46.7\% \pm 2.21$  and the average generalization error of NMF was  $42.5\% \pm 1.61$ . (Average chance generalization error was 58.9%.) The reduction in mean classification error was significant when PACA was compared to both PCA and NMF (Two-sample T-test<sup>8</sup>;  $p < 0.01$ ). Similarly significant improvements ( $p < 0.01$ ) were found for the case of high PACA regularization ( $\lambda = 0.1, \gamma = 0.1$ ), but in the case of low PACA regularization over topics ( $\lambda = 0.01$ ) there was only a significant improvement over PCA ( $p < 0.05$ ).

It is also apparent from the figures that there appears to be a range of values of  $K$  near the peak performance range for which the differences between PACA, NMF, and PCA are the largest. If we apply the two-sample T-test to data pooled for each value of  $K$  individ-

---

<sup>8</sup>Null hypothesis: the mean classification errors of the two compared algorithms are the same.

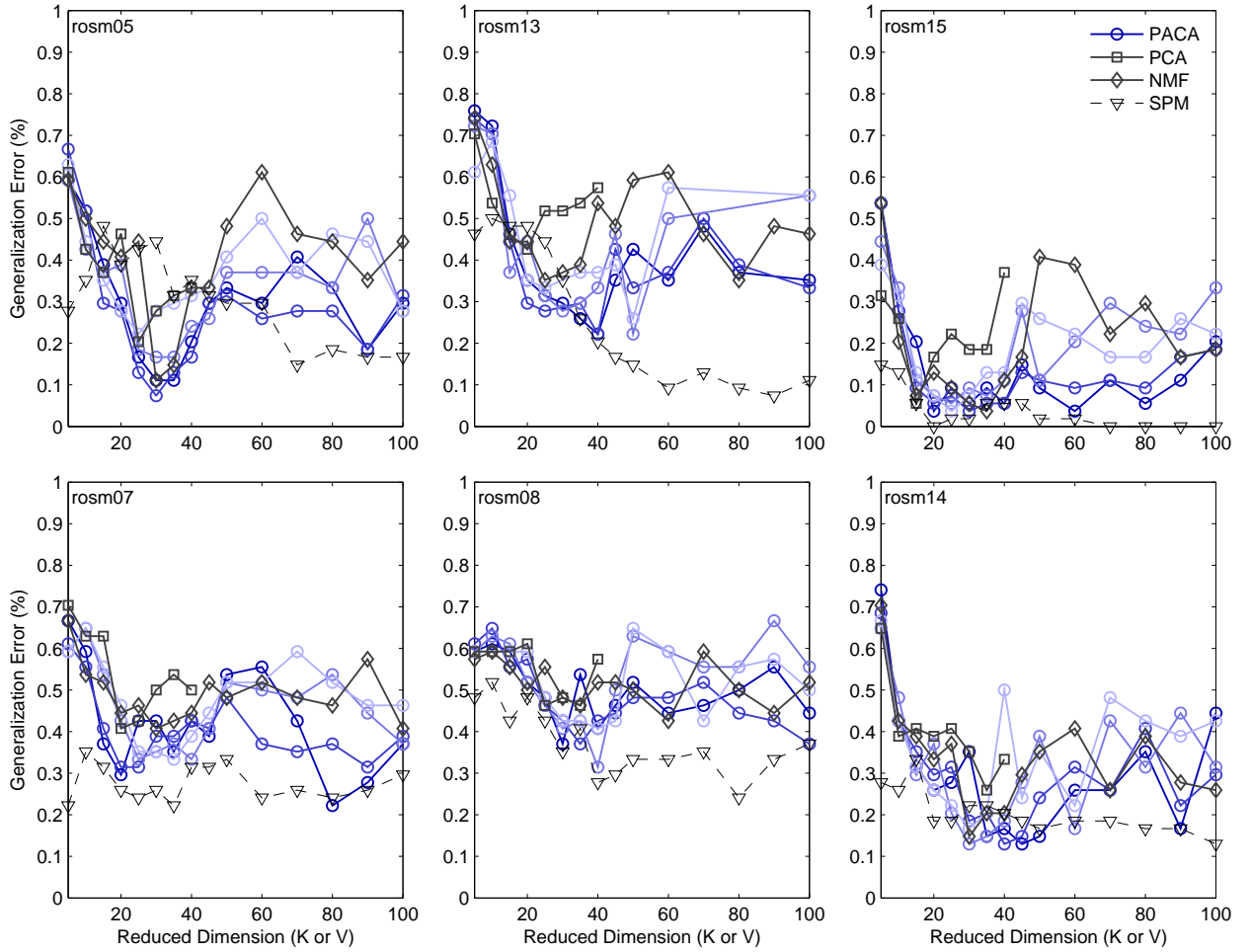


Figure 3.11: Cross validated classifier prediction error on the ROSM dataset. PACA (circles), PCA (squares), NMF (diamonds), and SPM (dotted triangles) were each run with  $K$  ranging from 5 to 100 on six subjects. PACA was run with four settings of  $(\lambda, \gamma)$ :  $(0.01, 0.01)$  (lightest blue),  $(0.01, 0.1)$  (light blue),  $(0.1, 0.01)$  (blue),  $(0.1, 0.1)$  (dark blue). A logistic regression classifier was trained on the reduced data to predict cognitive labels for each round of cross validation. The measured performance metric is the average error rate of the classifier on the test set across all rounds.

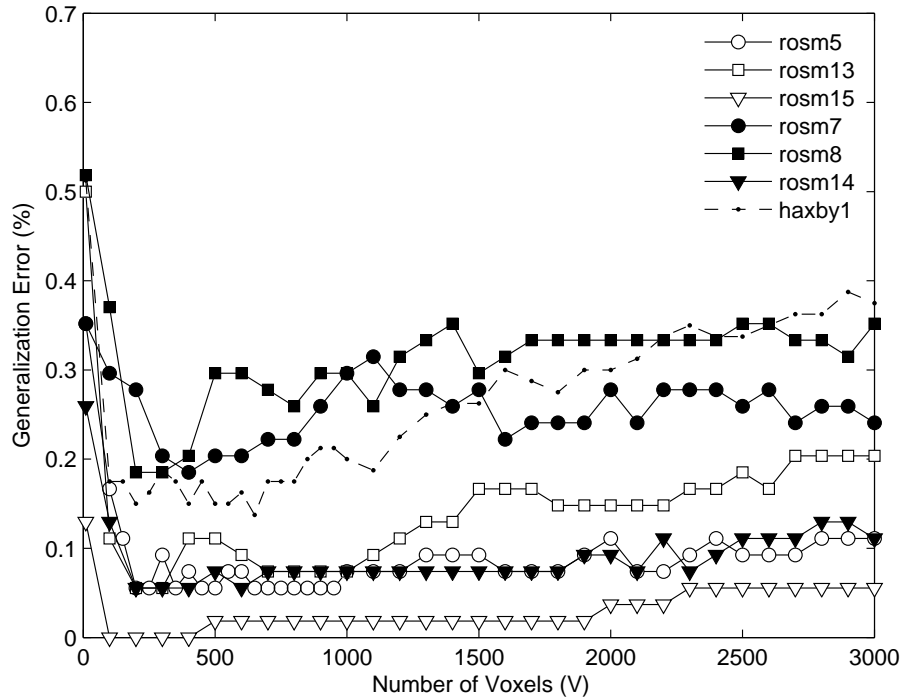


Figure 3.12: Cross validated classifier error for the SPM method, over larger sets of voxels, and for all subjects in the analysis. Cross validation experiments were run using SPM as the feature selection step with  $V$  ranging from 100 to 3000 in increments of 100. Plotted is the average generalization error of the classifier across all rounds of cross validation. The broad “U” shape of most of the curves indicates the optimal number of voxels to be included in the analysis.

ually, we find that PACA’s improvement in generalization performance is only significant for individual values of  $K$  in the range  $K = [40, 60]$  ( $p < 0.05$ ), and only for the cases where  $\lambda = 0.1$ .

In addition to the comparison of dimensionality reduction methods, we analyzed the performance of SPM across a wide range of values of  $V$  to obtain a baseline standard classification performance for each subject and to compared unsupervised and supervised feature selection methods generally. The classification error of the SPM method with  $V$  ranging from 100 to 3000 is shown for each subject used in our analysis in figure 3.12. The analysis of SPM was deemed necessary to determine to what extent each subject was “classifiable,” as three of the ROSM subjects (open dots) were known to have better generalization from the study phase of the ROSM experiment to the test phase than the

others (Robison, personal communication). These subjects appear qualitatively “easier,” with less classifier error across the range of included voxels. However, in every case, not just in the “easy” subjects, the minimum classifier error for SPM across all values of  $V$  was far below that of any of the unsupervised methods. The mean peak classifier error across all SPM values was  $9.63\% \pm 2.75$ , with a average improvement over PACA of  $9.82\% \pm 2.93$  ( $p < 0.02$ ).

### 3.3.4 Determining “Best Case” Improvements with Synthetic Data

Although PACA was unable to outperform SPM in the experimental datasets, we wished to investigate whether there was *any* case in which utilizing an unsupervised algorithm could do as well or better than supervised methods. To assess whether PACA could obtain a theoretical improvement over SPM in any case, we ran NMF, PCA, and PACA on the “ideal” synthetic dataset (section 3.2.3) with  $K$  ranging from 5 to 100 (figure 3.13. Figure 3.13 is labeled according to the same scheme used in the previous classifier generalization error in figures 3.10 and 3.11.

Unlike in the previous cases, there was a large and significant reduction in classification error when using PACA compared to every other feature selection or dimensionality reduction methods. However, these improvements were limited to the cases where  $\lambda = 0.1$ . For the high regularization case ( $\lambda = 0.1, \gamma = 0.1$ ), PACA had a mean improvement of  $10.9\% \pm 4.38$  over PCA (Two-tailed T-test<sup>9</sup>;  $p < 0.03$ ),  $8.83\% \pm 2.51$  over NMF ( $p < 0.005$ ), and a mean improvement of  $6.17\% \pm 1.98$  over SPM ( $p < 0.01$ ) when matched for individual settings of  $K$ . In the medium regularization case ( $\lambda = 0.1, \gamma = 0.01$ ), PACA had a mean improvement of  $8.75\% \pm 4.00$  over PCA ( $p < 0.05$ ) and  $5.33\% \pm 1.65$  over NMF ( $p < 0.01$ ), but no significant improvement over SPM. In other settings of PACA hyperparameters, there was no significant improvement between PACA and any other algorithm.

---

<sup>9</sup>Null hypothesis: the mean difference between classifier prediction errors of the two compared algorithms when matched for individual settings of  $K$  is zero.

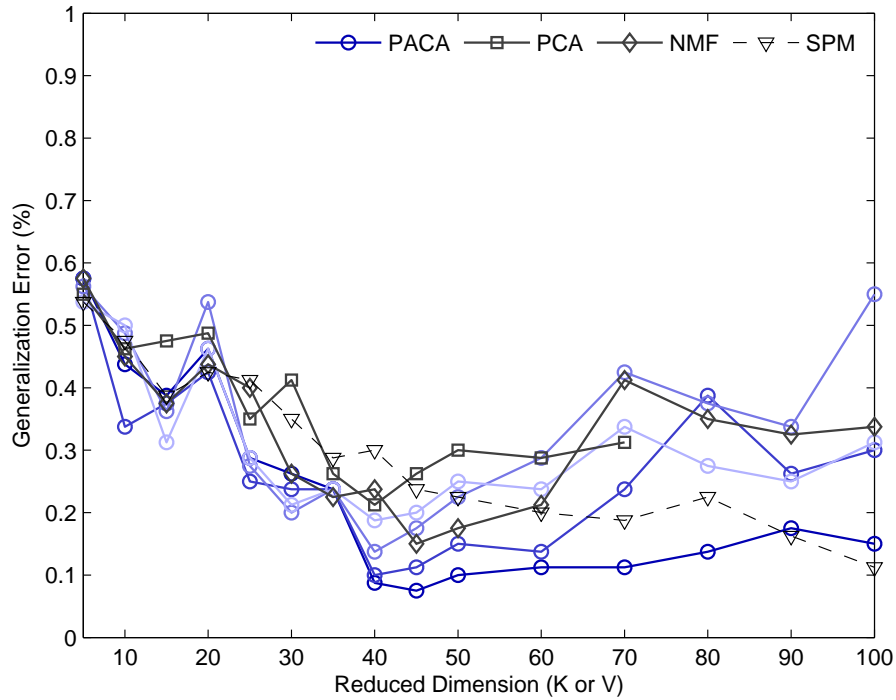


Figure 3.13: Classification performance on the “ideal” dataset.

## 3.4 Visualizing the Reduced Data

Once we had established that the PACA model was a practical improvement over existing means of dimensionality reduction, we attempted to find ways to visualize the neural topics found by the model and assess latent variables  $\mu$  and  $Z$  from a neuroscientific perspective.

### 3.4.1 Neural Topics

To visualize the neural topics discovered by PACA, we selected the model fitting results that had the least classifier prediction error from the single subject in the Haxby et al. (2001) dataset and subject 13 from the ROSM datasets. Both of these corresponded to the case where  $K = 50$ , although different hyperparameters were used in each case. To determine which of the 50 neural topics to visualize, we performed a uni-variate logistic regression for each of the  $K$  neural topics on each of the cognitive labels of the experiment using the

Matlab statistics toolbox. For each of the cognitive labels, we discarded non-significant ( $p \geq 0.05$ ) topics and sorted the remaining topics by the absolute value of the discovered relationship,  $|\beta_k|$ . Very few of the neural topics had uni-variate significant relationships for the cognitive labels, as measured through logistic regression.

Four neural topics with significant predictive relationships for four selected cognitive labels from the Haxby et al. (2001) experiment are shown in figure 3.14. Each subplot shows the values of  $\mu_{k,v}$  for four representative axial slices; negative values of  $\mu_{k,v}$  are shown in blue, values near zero in red, and positive values are shown in yellow. The color scales of each neural topic  $\mu_k$  are normalized for that topic. The value of  $\beta_k$  is indicated for each visualized  $\mu_k$ . A similar plot showing two neural topics for each of the cognitive labels in the ROSM experiment is presented in figure 3.15

Although a rigorous neuroscientific analysis of the discovered neural topics is out of the scope of this thesis, it's interesting to note the varying degrees of smoothness and spatial organization of the different topics. For instance, neural topics 43 and 44 in figure 3.14 both show clear “pockets” of excitation in somewhat similar areas, although topic 44 has a positive  $\beta$  for the “house” category while topic 43 has a negative  $\beta$  for the “cat” category. Thus, these discovered topics are qualitatively consistent with the distributed and overlapping representations of objects in ventral temporal cortex discussed in Haxby et al. (2001). Topographical organization of the ROSM topics in frontal cortex is much less clear (figure 3.15), but there no evidence yet to suggest for or against spatially localized representations of retrieval orientation.

### 3.4.2 Brain States

To gain a qualitative notion of the differences of the reduced data found by the different dimensionality reduction algorithms, we visualized the  $\mathbf{Z}$  matrices of the PACA models used in the previous section. To form the plots in in figures 3.16 and 3.17, we created an image where the relative brightness of each pixel represents the value  $z_{k,t}$ . We then sorted

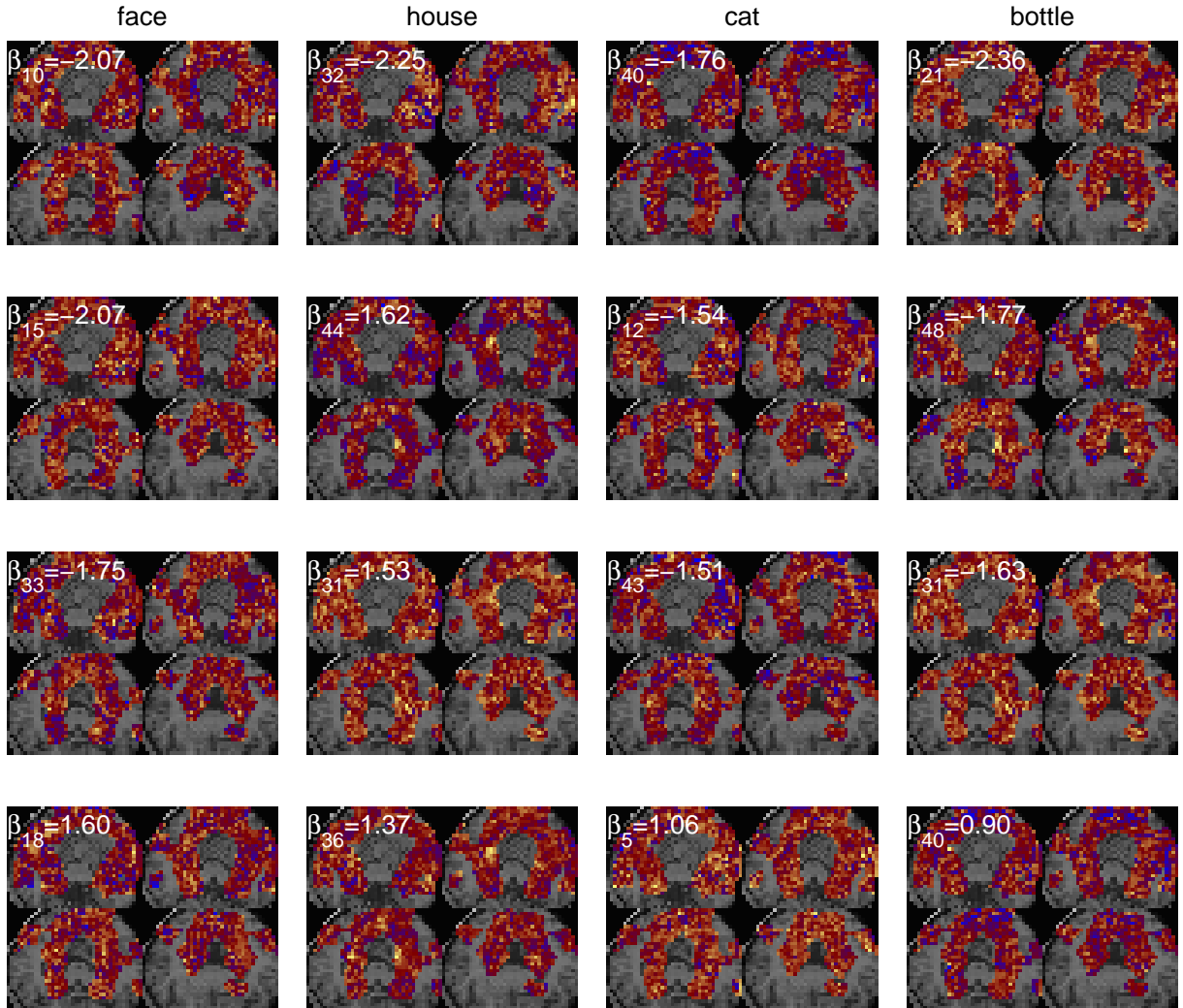


Figure 3.14: Visualization of selected neural “topics” discovered by PACA on the Haxby et al. (2001) dataset ( $\lambda = 0.1, \gamma = 0.01, K = 50$ ). Uni-variate logistic regression was performed to establish the strength of a predictive relationship between each of neural topics and each cognitive label. Shown are four significant topics for four selected cognitive categories: face, house, cat, and bottle. The images are four axial slices of cortex with the values of  $\mu_{k,v}$  superimposed; blue colors indicate negative values of  $\mu_{k,v}$ , while yellow values indicate positive values. Color scales have been normalized for each topic. The strength of the predictive relationship  $\beta_k$  is also displayed for each topic.

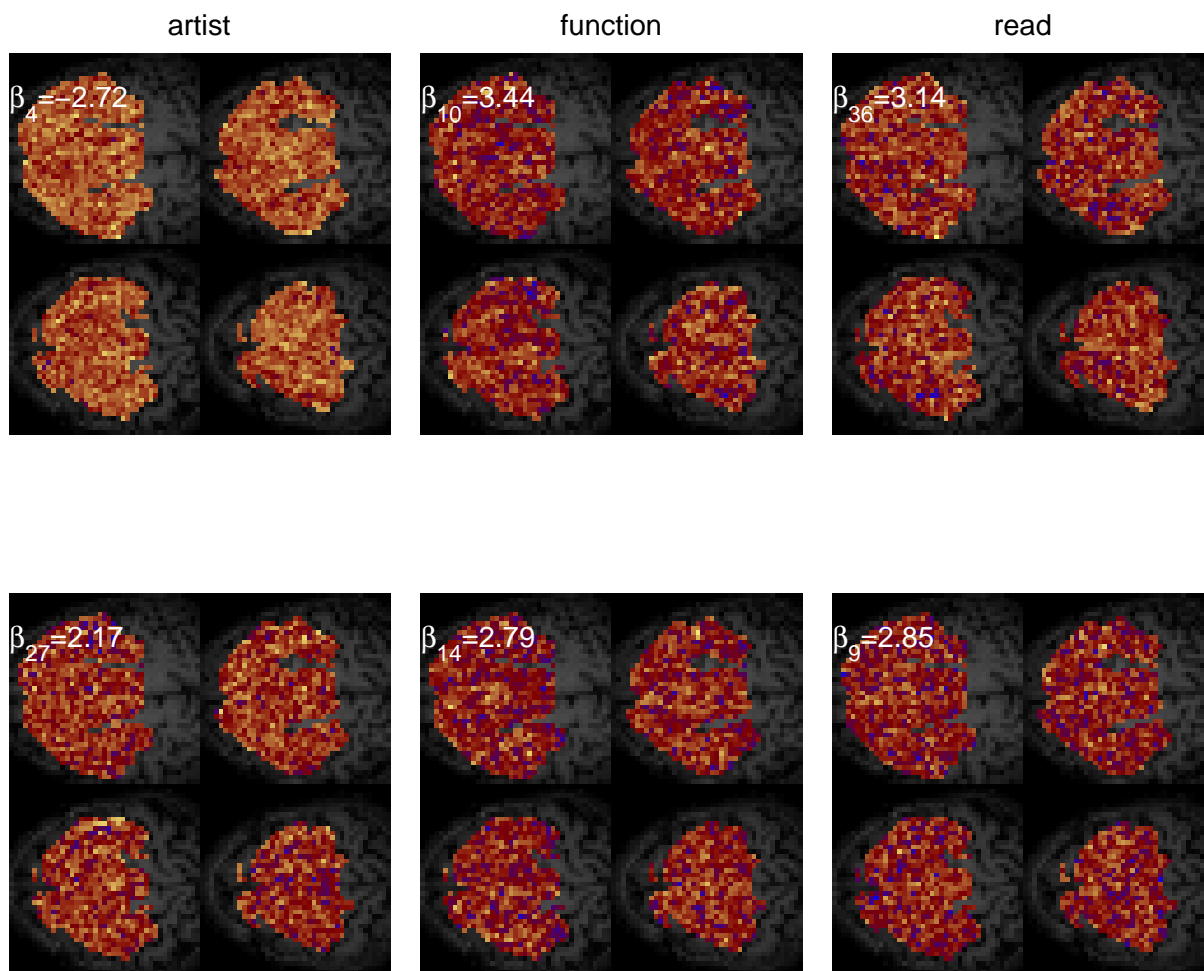


Figure 3.15: Visualization of selected neural “topics” discovered by PACA on subject 13 from the ROSM dataset ( $\lambda = 0.01, \gamma = 0.1, K = 50$ ). Uni-variate logistic regression was performed to establish the strength of a predictive relationship between each of neural topics and each cognitive label. Shown are two significant topics for the three retrieval orientations: artist, function, and read. The images are four axial slices of cortex with the values of  $\mu_{k,v}$  superimposed; blue colors indicate negative values of  $\mu_{k,v}$ , yellow values indicate positive values. Color scales have been normalized for each topic. The strength of the predictive relationship  $\beta_k$  is also displayed for each topic.

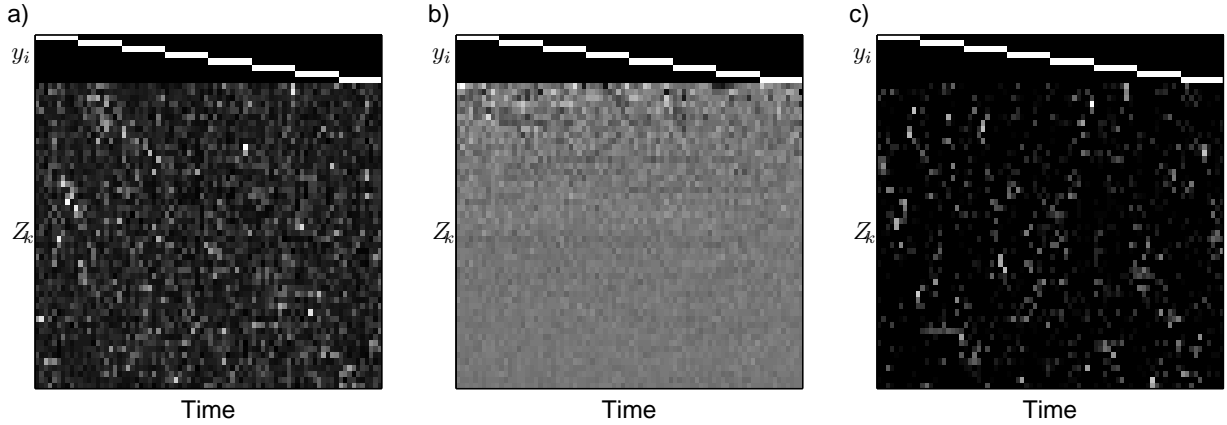


Figure 3.16: The “Z-Images” of the Haxby et al. (2001) dataset from different dimensionality reduction algorithms: **a**, PACA, **b**, PCA, and **c**, NMF. Timepoints are grouped by experimental category and ordered within category by correlation based dendrogram. For non-PCA decompositions, topics are ordered by correlation based dendrogram; in **b**, topics are ordered according to decreasing eigenvalues. All shown for  $K = 50$ . The top 8 rows of each picture show the category groupings.

the columns of the image to group similar timepoints together, and sorted the rows of the image to group similar neural topics together, since the indexes  $k$  and  $t$  are exchangeable by assumption. Sorting was done by taking the ordering of a dendrogram produced through correlation-based agglomerative clustering along each of the respective dimensions. We then concatenated the sorted, pixel-based representation of  $\mathbf{Z}$  (“Z-image”) with the associated matrix of binary cognitive labels  $\mathbf{Y}$ , so that the reduced representation of the different cognitive states could be directly and visually compared. In the Z-image, a neural topic that is associated with a particular cognitive state will appear as a dark or light horizontal bar, while groups of such topics will appear as a dark or light block.

Qualitatively, we see that the three dimensionality reduction algorithms-PACA, PCA, and NMF-produced dramatically different representations of the observed data in the reduced space of  $\mathbf{Z}$ . As expected, the encoding of voxel activity found by PACA is similar to the encoding of faces discussed in section 3.1: most time points are represented by many small values of  $z_{k,t}$ , with a few very large values for each timepoint. The Z-image of NMF appears qualitatively similar to that of PCA, except that it is far more sparse. PCA, on the other hand, has a dramatically asymmetrical representation, as most of the variance in the

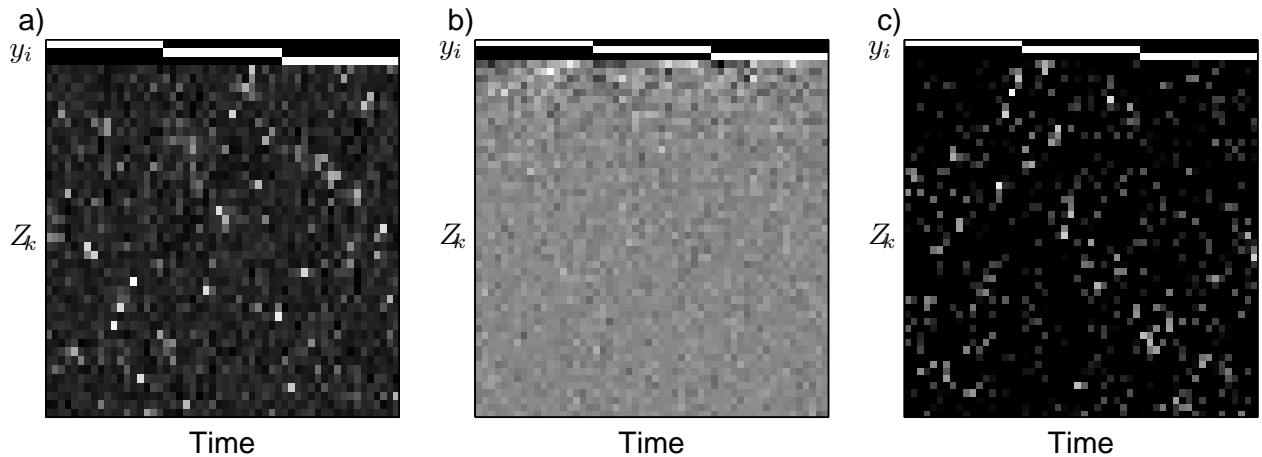


Figure 3.17: The “Z-Images” of subject 13 of the ROSM dataset from different dimensionality reduction algorithms: **a**, PACA, **b**, PCA, and **c**, NMF. Timepoints are grouped by experimental category and ordered within category by correlation based dendrogram. For non-PCA decompositions, topics are ordered by correlation based dendrogram; in **b**, topics are ordered according to decreasing eigenvalues. All shown for  $K = 50$ . The top 3 rows of each picture show the category groupings.

Z-image is accounted for by the first few principal components. Generally, very few neural topics appear visually to be strongly associated with any particular topic, which agrees with the result from the previous section that very few PACA topics were significantly predictive of any cognitive label through logistic regression.

# Chapter 4

## Discussion

In this thesis, we have developed four central points. First, we showed how a set of neuroscientifically motivated assumptions could be instantiated in a probabilistic model by representing that model graphically and defining conditional probability distributions (section 2.1). Second, we compared these modeling assumptions to those behind two currently popular methods of dimensionality reduction, PCA and NMF. We found that PCA and NMF did not satisfy the neuroscientific assumptions behind the new PACA model (section 2.5). Third, we showed experimentally that the PACA model is as good or better than the other models at decomposing functional imaging data into matrix factors (section 3.3.2), and that classifying on PACA reduced data yields better classification accuracy than classifying on reduced data from PCA or NMF (section 3.3.3). Furthermore, in certain synthetic conditions, the PACA model boosted classification performance beyond that obtained by supervised methods (section 3.3.4). Finally, we began to explore the qualitative properties of the discovered neural topics and brain states in section 3.4, and we found general agreement with earlier work using the same subjects.

## 4.1 Key Points

### 4.1.1 Modeling Assumptions are Important

The single most important implication of these results can be summed up very simply: the intrinsic assumptions behind any method of dimensionality reduction have a crucial impact on the resulting analysis, whether or not these assumptions are explicitly expressed. Equivalently, we can improve the quality of our fMRI analysis experiments by incorporating existing neuroscientific knowledge into our data mining techniques. Furthermore, we can accomplish this by explicitly specifying conditional probability distributions in a generative model of the data we are trying to analyze. Thus, since there are so many methods of dimensionality reduction (McKeown et al., 1998; Shen & Meyer, 2006; Fan et al., 2006), qualitative comparison of the underlying assumptions might provide one means of determining which methods are appropriate for fMRI analysis and which methods are not. When the potential models can all be represented graphically, qualitative comparisons becomes intuitive and easy; in this thesis, we were able to compare techniques for which the underlying generative assumptions were not immediately clear (Jolliffe, 2002; Tipping & Bishop, 1999; Lee & Seung, 1999).

However, it is also important to recognize that comparing the generative assumptions underlying dimensionality reduction methods is only useful when it is possible to make definitive judgments about the suitability of those assumption. In this thesis, we motivate the PACA modeling assumptions from a neuroscientific perspective, but the assumptions we made are relatively straightforward and uncontroversial. If we wish to expand the complexity of the model to better reflect the nuances of the data, we will need to increase the specificity of the underlying assumptions. Presumably, at some point the assumptions will become purely speculative, and the utility of qualitatively comparing such complex and speculative models will be much less than in the basic case examined in this thesis.

### 4.1.2 Supervised vs. Unsupervised Analysis

This thesis also provides an investigation into the practical benefit of unsupervised dimensionality reduction techniques as an alternative to supervised feature selection methods, such as SPM. In this case, we found that no unsupervised method was as effective at reducing the dimensionality of the data as the supervised SPM method. Nonetheless, in the “ideal” synthetic case, PACA was able to boost classification accuracy beyond that of supervised methods. This result supports the intuitive notion that neither purely supervised nor purely unsupervised methods are optimal for fMRI analysis; while SPM ignores structural information in the voxel data and considers each voxel univariately, PACA completely ignores all information about cognitive state that the behavioral cognitive labels provide.

One potential “hack” to combine supervised and unsupervised analysis is simply to run the dimensionality reduction algorithm PACA on voxels that have been preselected by SPM to be informative. This method would essentially replicate the “ideal” synthetic case, but without the “peeking” at the entire dataset. The difficulty with such a solution is that the computational requirements will be very demanding due to the need to run dimensionality reduction on each iteration of cross validation, and this solution would sacrifice all other benefits of unsupervised analysis in the name of reduced classification error. Furthermore, within the framework of probabilistic generative models, it is possible to account for both labeled and unlabeled in a rigorous fashion (Nigam et al., 2000).

### 4.1.3 Choosing the Right Number of Components

One fundamental issue critical to any dimensionality reduction or latent variable model is the question of how to select  $K$ , the number of components in the reduced dimensions of the model. Interestingly, there appears to be a relatively narrow range of  $K$  for which both reconstruction error and classification prediction error are minimized most efficiently, and this range is consistent across both the Haxby et al. (2001) dataset and the ROSM datasets. In figures 3.8 and 3.9, the reconstruction error for PACA across all parameter

sets consistently levels off at approximately  $K = 40$ . Thus,  $K = 40$  is the approximately the number of components at which the PACA model can reproduce the original voxel data most effectively. Similarly, as discussed in section 3.3.3, a consistent minimum of classification error occurred approximately in the range  $K = [30, 50]$ .

Because the reconstruction error is a measure that is computed without looking at the labels of the data, it is therefore potentially possible to maximize the degree to which the PACA reduced data predicts cognitive labels without any supervised experimentation at all. One could choose  $K$  by calculating generalization reconstruction error for a range of values and then selecting the value closest to the “elbow” of the reconstruction error curve. Such a procedure would allow non-trivial choices of  $K$  for exploratory “mind-reading” experiments in which cognitive task labels are entirely absent. For example, Hasson et al. (2004) examined multiple subject fMRI responses to clips of the movie *The Good, the Bad, and the Ugly*, and used reverse-correlation to discover which cognitive aspects of the movie scenes the subjects’ brains were responding to. A similar analysis could be repeated using a dimensionality reduction technique such as PACA, and finding those timepoints in the movie that were represented by similar brain states in the reduced space—but only if there is an unsupervised means of choosing hyperparameters that also optimizes “mind-reading” performance.

#### **4.1.4 Benefits of Regularization**

The experimental results in this thesis also reinforce the importance of regularization in overspecified problems, such as the “mind-reading” problem. As noted in sections 3.3.3, PACA tended to minimize classifier error when PACA was run with relatively high regularization hyperparameters ( $\lambda = 0.1$  and/or  $\gamma = 0.1$ ). The sensitivity of PACA to particular settings of the hyperparameters was less important for the reconstruction error paradigm, but NMF was notably very sensitive to overfitting. We also note that the benefit of regularization increased as  $K$  increased, indicative of the propensity of the model to overfit as the

complexity of the model increased. Thus, the use of prior probabilities over the parameters in probabilistic generative models is successful in preventing overfitting.

#### **4.1.5 Limitations of Experimental Results**

There are several important limitations of the experimental results in this thesis that arise due to the averaging step of the analysis (figure 3.6c). First, one of the goals of most “mind-reading” experiments, including the analysis of Robison et al. (2007), is to predict cognitive state at the maximum temporal resolution of fMRI. By averaging together timepoints, we bring the data into line with our model’s assumptions of exchangeability among timepoints, but we also lose the ability to assess experimentally the practical use of dimensionality reduction algorithms in these more detailed analyses. Furthermore, averaging together experimental mini-blocks increases the signal-to-noise ratio of the dataset. Higher signal-to-noise ratios results in much more accurate classifiers than in the more general case of non-averaged time-series data. Specifically, we suspect that utility of SPM as a method for feature selection might be partially inflated due to the lower amounts of noise; with less noise, it should be easier to tell which voxels are informative about the cognitive labels and which are not. Regularization should also be more important when the noise content of the data is high.

Another limitation of the current experiment is that we were limited by time constraints when exploring settings of the hyperparameters  $\lambda$  and  $\gamma$ . We noted earlier that the “high” regularization condition produced the lowest reconstruction and classifier errors; ideally, we would be able to map out performance over a large portion of the hyperparameter space, to determine the size and uniqueness of the optimal set of hyperparameters. Thus, the experimental results presented in this thesis do not represent the optimal performance of the PACA model.

## 4.2 Directions for Future Work

There are several ways in which the PACA model could be improved, both analytically and practically. First, one could substitute the Gaussian prior over  $\mu$  and corresponding L2 regularization with a Laplace distribution to enforce L1 regularization over the neural topics. As discussed earlier (section 2.4.3), regularization using the L1 norm causes the model to find sparse solutions to the problem; if we used a L1 norm over the neural topics, PACA would find neural topics that were sparsely distributed in space over cortex. Sparse neural topics might be more desirable than the “smooth” topics found in the current PACA model with L2 regularization; although Haxby et al. (2001) found distributed representations of objects in ventral temporal cortex, there is nonetheless a certain amount of topographical organization inherent in much of cortical structure. Furthermore, the improvement of PACA over supervised methods on the “ideal” synthetic dataset might suggest that dimensionality reduction works best when the number of voxels in the neural topics is relatively small and the distribution is sparse. Using a sparse spatial prior distribution might therefore be a way to approximate the benefit of the voxel selection used to create the “ideal” dataset originally in an unsupervised manner.

A second potential improvement to the existing model would be to use a more powerful approximate inference technique to infer the posterior *distribution* of the parameters,

$$p(\mu, \mathbf{Z} | \mathbf{X}, \eta) = p(\mathbf{X} | \mu, \mathbf{Z}, \eta) p(\mu | \eta) p(\mathbf{Z} | \eta), \quad (4.1)$$

rather than finding the MAP estimate  $\hat{\mu}_{\text{MAP}}, \hat{\mathbf{Z}}_{\text{MAP}}$ . If the posterior distribution is known, we can examine the posterior directly to find whether or not the posterior distribution is multi-modal and to estimate the confidence intervals for particular parameters. We can then easily and more rigorously generalize our analysis to new data using the same inference process as was used to find the original data, simply by using our posterior distribution on one step as the prior distribution on the next step (a process known as a “Bayesian update” (Russell

& Norvig, 2003)). One promising means of performing efficient and accurate approximate inference is *variational inference* (Blei et al., 2003).

Finally, there are a number of ways the PACA graphical model could be expanded structurally to remove the limiting, unrealistic assumptions of exchangeability in both space and time. To account for time-series data, a dependency could be introduced between the brain state at time  $t$  and the brain state at previous times  $t - 1, \dots, t - d$ . Similarly, to account for the spatial relationship between voxels, the topic mean of a given voxel could be modeled to depend on that of its spatially located neighbors. In this manner, the PACA model can be expanded in the future to account for more types of data.

# References

- Andersen, A. H., Gash, D. M., & Avison, M. J. (1999). Principal component analysis of the dynamic response measured by fmri: A generalized linear systems framework. *Magnetic Resonance Imaging*, 17(6), 795–815.
- Battle, A., Chechik, G., & Koller, D. (2006). Temporal and cross-subject probabilistic models for fmri prediction tasks. Technical report, Stanford University.
- Blei, D., Ng, A. Y., & Jordan, M. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993–1022.
- Chigirev, D. & Stephens, G. (2006). Predicting base features with supervoxels. Technical report, Princeton University.
- Cox, D. D. & Savoy, R. L. (2003). Functional magnetic resonance imaging (fmri) "brain reading": detecting and classifying distributed patterns of fmri activity in human visual cortex. *NeuroImage*, 19, 261–270.
- Davatzikos, C., Ruparel, K., Fan, Y., Shen, D., Acharyya, M., Loughead, J., Gur, R., & Langleben, D. (2005). Classifying spatial patterns of brain activity with machine learning methods: Application to lie detection. *Neuroimage*, 28, 663–668.
- Detre, G., Polyn, S.M., Moore, C.D., Natu, V.S., Singer, B.D., Cohen, J.D., Haxby, J.V., Norman, & K.A. (2006). The multi-voxel pattern analysis (mvp) toolbox. Poster presented at the Annual Meeting of the Organization for Human Brain Mapping.
- Fan, Y., Shen, D., , & Davatzikos, C. (2006). Detecting cognitive states from fmri images by machine learning and multivariate classification. *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop*.
- Fletcher, R. (1987). *Practical Methods of Optimization* (Second ed.). Wiley.
- Ford, J., Farid, H., Makedon, F., Flashman, L. A., McAllister, T. W., Megalooikonomou, V., & Saykin, A. J. (2003). Patient classification of fmri activation maps. *MICCAI*, 2879, 58–65.
- Formisano, E., Esposito, F., Salicrú, F. D., & Goebel, R. (2004). Cortex-based independent component analysis of fmri time series. *Magnetic Resonance Imaging*, 22, 1493–1504.

- Frank, L., Buxton, R., & Wong, E. (1998). Probabilistic analysis of functional magnetic resonance imaging data. *Magn Reson Med*.
- Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Booth, M., & Rossi, F. (2006). *GNU Scientific Library Reference Manual* (Revised Second ed.). Bristol, UK: Network Theory Limited.
- Hasson, U., Nir, Y., Levy, I., Fuhrmann, G., & Malach, R. (2004). Intersubject Synchronization of Cortical Activity During Natural Vision. *Science*, 303(5664), 1634–1640.
- Hastie, T., Tibshirani, R., & Friedman, J. H. (2003). *The Elements of Statistical Learning* (First ed.). Springer.
- Haxby, J. V., Gobbini, M. I., Furey, M. L., Ishai, A., Schouten, J. L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science*, 293.
- Haynes, J. D. & Rees, G. (2005). Predicting the stream of consciousness from activity in human visual cortex. *Current Biology*, 15, 1301–1307.
- Haynes, J. D. & Rees, G. (2006). Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7.
- Jolliffe, I. (2002). *Principal component analysis* (Second ed.). New York: Springer. Online Edition.
- Kriegeskorte, N., Goebel, R., & Bandettini, P. (2006). Information-based functional brain mapping. *PNAS*, 103(10), 3863–3868.
- Lee, D. D. & Seung, S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401, 788–791.
- Lee, D. D. & Seung, S. (2001). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13, 556–562.
- Lin, C. J. (2007). Projected gradient methods for non-negative matrix factorization. *Neural Computation*.
- McKeown, M. J., Makeig, S., Brown, G. G., Jung, T.-P., Kindermann, S. S., & Sejnowski, A. J. B. T. J. (1998). Analysis of fmri data by blind separation into independent spatial components. *Human Brain Mapping*, 6, 160–188.
- McKiernan, K. A., Kaufman, J. N., Kucera-Thompson, J., & Binder, J. R. (2003). A parametric manipulation of factors affecting task-induced deactivation in functional neuroimaging. *Journal of Cognitive Neuroscience*, 15(3), 394–408.
- Mitchell, T., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., & Newman, S. (2004). Learning to decode cognitive states from brain images. *Machine Learning*, 57, 145–175.

- Mitchell, T. & Rustandi, I. (2006). Hidden Process Models. *Proceedings of the International Conference on Machine Learning*.
- Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39(2), 103–134.
- Norman, K. A., Polyn, S. M., Detre, G. J., & Haxby, J. V. (2006). Beyond mind-reading: multi-voxel pattern analysis of fmri data. *TRENDS in Cognitive Sciences*, 10(9), 424–430.
- Paul Komarek, A. M. (2005). Making logistic regression a core data mining tool: A practical investigation of accuracy, speed, and simplicity. technical report.
- Petersson1, K. M., Nichols, T. E., Poline, J.-B., , & Holmes, A. P. (1999). Statistical limitations in functional neuroimaging i. non-inferential methods and statistical models. *Phil. Trans. R. Soc. Lond. B*, 354(1239-1260).
- Polyn, S. M., Natu, V. S., Cohen, J. D., & Norman, K. A. (2005). Category-specific cortical activity precedes retrieval during memory search. *Science*, 310, 1963–1966.
- Robison, S., Frankel, H., & Norman, K. (2007). Classifying retrieval orientation states: Dissecting retrieval dynamics underlying previously studied items. Research summary, work in progress.
- Roweis, S. & Ghahramani, Z. (1999). A Unifying Review of Linear Gaussian Models. *Neural Computation*, 11(2), 305–345.
- Russell, S. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (Second ed.). Pearson Education, Inc.
- Shen, X. & Meyer, F. G. (2006). Nonlinear dimension reduction and activation detection for fmri dataset. In *Conference on Computer Vision and Pattern Recognition Workshop*.
- Tipping, M. E. & Bishop, C. M. (1999). Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society, Series B*, 61(3), 611.
- Wang, X., Tian, J., Li, X., Dai, J., & Ai, L. (2004). Detecting brain activations by constrained non-negative matrix factorization from task-related BOLD fMRI. In Amini, A. A. & Manduca, A. (Eds.), *Medical Imaging 2004: Physiology, Function, and Structure from Medical Images*, (pp. 675–682).