Predictive fMRI Analysis for Multiple Subjects and Multiple Studies (Thesis)

Indrayana Rustandi
Computer Science Department
Carnegie Mellon University

CMU-CS-10-117
May 11, 2010

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Tom M. Mitchell, Chair
Zoubin Ghahramani
Eric Xing
David Blei, Princeton University

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the W.M. Keck Foundation under grant number DT123107.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Contents

List of Tables

List of Figures

Abstract

In the context of predictive fMRI data analysis, the state of the art is to perform the analysis separately for each particular subject in a specific study. Given the nature of the fMRI data where there are many more features than instances, this kind of analysis might produce suboptimal predictive models since the data might not be sufficient to obtain accurate models. Based on findings in the cognitive neuroscience field, there is a reason to believe that data from other subjects and from different but similar studies exhibit similar patterns of activations, implying that there is some potential for increasing the data available to train the predictive models by analyzing together data coming from multiple subjects and multiple studies. However, each subject's brain might still exhibit some variations in the activations compared to other subjects' brains, based on factors such as differences in anatomy, experience, or environment. A major challenge in doing predictive analysis of fMRI data from multiple subjects and multiple studies is having a model that can effectively account for these variations.

In this thesis, we propose two classes of methods for predictive fMRI analysis across multiple subjects and studies. The first class of methods are based on the hierarchical linear model where we assume that different subjects (studies) can have different but still similar parameter values. However, this class of methods are still too restrictive in the sense that they require that the different fMRI datasets to be registered to a common brain, a step that might introduce distortions in the data. To remove this restriction, we propose a second class of methods based on the idea of common factors present in different subjects/studies fMRI data. We consider learning these factors using principal components analysis and canonical correlation analysis. Based on the application of these methods in the context two kinds of predictive tasks—predicting the cognitive states associated with some brain activations and predicting the brain activations associated with some cognitive states—we show that we can indeed effectively combine fMRI data from multiple subjects and multiple studies and obtain significantly better accuracies compared to single-subject predictive models.

Acknowledgments

Many thanks first and foremost to Tom Mitchell for his guidance and support over all these years. Tom gave countless valuable advice over the years, and he taught me the importance of having a good intuition. I will in particular remember his remark that being the first might be more important than being the smartest. Also thanks to Zoubin Ghahramani, Eric Xing, and Dave Blei for agreeing to serve on my thesis committee and giving lots of useful feedback on the thesis. I have enjoyed the interactions that I have had with past and present members of the Brain Image Analysis Research and the CCBI groups, including Marcel Just, Sandesh Aryal, Kai-min Chang, Vlad Cherkassky, Khalid El-Arini, Rebecca Hutchinson, Rob Mason, Mark Palatucci, Dean Pomerleau, and Svetlana Shinkareva. I have also had fruitful discussions with visitors that we have had over the years, including Luis Barrios, Avi Bernstein, Russ Greiner, John-Dylan Haynes, Danny Silver, and Stephen Strother. Although my interaction with Jay Kadane was brief, he gave a valuable suggestion on how to evaluate when two accuracies are significantly different using the jackknife procedure. The thesis research became tremendously more productive in the beginning of 2009 when I finally had working access to Star-P running on pople, a 768-core SGI Altix machine at the Pittsburgh Supercomputing Center (PSC); thanks to Joel Welling for facilitating access to pople, and to Raghu Reddy for all his help early on resolving issues with Star-P and for being a liaison to the Star-P developers. I would also like to recognize members of the speaking skills committee, especially Scott Fahlman, for their constant feedback and criticisms have made me a better speaker. Last but not least, the outstanding support staff also deserve a special mention, especially Sharon Burks, Deb Cavlovich, Sharon Cavlovich, and Catherine Copetas. Life as a graduate student would have been a lot harder without their support.

Buat Mamah dan Papah, for Wendy, and AMDG.

Chapter 1

Introduction

Functional Magnetic Resonance Imaging (fMRI) is a non-invasive technique to capture brain activations with a relatively high spatial resolution, in the order of several cubic millimeters. fMRI presents an opportunity to advance our understanding in how the brain works, but it also presents a challenge as to how to extract the information present in the data. It has been shown (reviewed in section 1.2.1) that machine learning techniques, including classification and regression methods, can answer this challenge. In particular, there are numerous results showing that when these techniques can yield significantly better-than-random predictive accuracies when applied to fMRI data.

Despite the success, there are fundamental limitations of existing machine learning approaches when used to analyze fMRI data. In a typical fMRI study, there are in most cases more than one subject, each subject having a unique brain in terms of shape and size. In machine learning terminology, the feature space is different for different subjects, making it problematic to train a classifier using data from multiple subjects. This problem can be alleviated by registering all the subjects' brain to a common space. However, the uncertainty introduced by the registration is often ignored. Furthermore, it might still be the case that the patterns of activations for the same cognitive process might be different for different subjects because of the influence of factors such as each subject's unique experience and differences in the brain's vascular density for different subjects. As such, currently machine learning techniques are often applied independently for each individual subject, or they are applied to normalized data for all the subjects without properly accounting for the uncertainty introduced by the normalization process and the inter-subject variations still present in the data.

In more than a few cases, there have been more than one fMRI study done to study a common cognitive phenomenon. For instance, there have been a couple of fMRI experiments done in our group to study semantic representations in the brain, with one study using words as the stimuli while pictures were used in the other study. In some cases, a research group runs the same study multiple times; an example is a study described in Wei et al. (2004), in which the fMRI activations corresponding to the auditory 2-back task were captured over eight sessions, with at least 3 weeks of time in between sessions. In other cases, several different research groups run similar studies, for instance a study described in Casey et al. (1998), in which an fMRI study on spatial working memory was done at four different institutions. Intuitively, there is some information common across these studies, mixed with variations introduced by, among others, different experimental conditions and different stimulus types. Current machine learning approaches are not flexible enough to handle these variations, so they are usually applied independently for each individual study, even for studies with the same subjects.

With that consideration, the main thesis is

It is possible to invent machine learning and statistical techniques that can combine data from multiple subjects and studies to improve predictive performance, such that common patterns of activations can be distinguished from subject-specific and/or study-specific patterns of activations.

In other words, despite the challenges outlined above, in this thesis we show that we can develop methods that can account for data from multiple subjects and studies. These methods are measured by

their ability to make optimal predictions of some quantity of interest (e.g. class in classification) when presented with previously unobserved data, conditioned on the data that have been observed. In addition, the methods can also reveal common patterns vs patterns that are specific to specific subjects or studies.

What would be the benefits of being able to combine data across multiple domains in the case of fMRI data? fMRI data is high-dimensional but very few training examples relative to the number of dimensions, a not-so-ideal combination from the perspective of machine learning. By combining data across multiple subjects and studies, one benefit is that we can increase the number of training examples available, which can then improve the performance of these machine learning techniques. Another benefit would be the ability to extract components shared across one or several subjects and/or studies and distinguish them from components specific to specific subjects and/or studies. This is beneficial from the cognitive neuroscience perspective, because these methods can allow cognitive neuroscientists to integrate data from multiple subjects and studies so that common mechanisms for specific cognitive activities are revealed. By validating on their predictive performance, we can ascertain that these mechanisms can generalize to new instances, verifying that they are reliable and reproducible.

One fundamental assumption made in this thesis is that fMRI activations across subjects and studies are not completely independent. More formally, we assume that there are dependencies among the probability distributions of the different subjects and studies'fMRIactivations. If this assumption is violated, then there will not be any use in trying to leverage data from other subjects or studies, because the data for one subject or study does not provide any information at all about the data for another subject or another study. Nonetheless, as shown in this thesis, we can indeed obtain better predictive accuracies when integrating fMRI data across subjects and/or studies, indicating that this assumption holds to some extent.

One might alsothink that there is no purpose in integrating fMRI data across subjects and studies when we have infinite training data for all the subjects and all the studies. Indeed, when this is the case, we have a complete characterization of the uncertainties present in each subject's fMRI data, so there is no leverage provided by the other subjects' data. Nevertheless, methods that integrate fMRI data across subjects and studies still have value in this scenario because they can still reveal the similarities and differences that are present in the brain activations across subjects and/or studies.

The problem considered in this thesis can also be framed as an instance of the more general machine learning problem of methods that can be applied to multiple related tasks. In the context of the thesis, task refers to a particular subject and/or study. A review of existing work in machine learning for dealing with multiple related tasks is presented in section 1.2.3. A notable aspect present when considering multiple related tasks in the context of fMRI data is the fact that the feature space of each task is not necessarily the same as the feature space of another task. This is due to differences present in the brains and the brain activations across individuals.

Next, we present an overview of fMRI in section 1.1, and consider related works in section 1.2. We close this chapter by describing the fMRI datasets that are used in this thesis in section 1.3. In chapter 2, we consider incorporating the hierarchical linear models to extend predictive methods so that they can integrate fMRI data coming from multiple subjects and/or studies. Chapter 3 describes an alternative approach where we consider the commonalities present in the fMRI data across subjects and/or studies in terms of some higher-order factors. Results of applying the factor-based approach are described in two case studies, contained in chapters 4 and 5. We conclude and describe some possible directions for future work in chapter 6.

1.1 fMRI Overview

fMRI utilizes a strong magnetic field to detect fine-grained changes in the magnetic properties of the brain. In particular, fMRI is designed to take advantages of the changes in the magnetic properties of oxyhemoglobin and deoxyhemoglobin during neural activations compared to when neural activations are absent. Oxyhemoglobin (hemoglobin when it is carrying oxygen) is diamagnetic, while deoxyhemoglobin (hemoglobin when it is not carrying oxygen) is paramagnetic. At resting state, in the absence of any neural activations, there is a specific proportion between oxyhemoglobin and deoxyhemoglobin. When a neuron

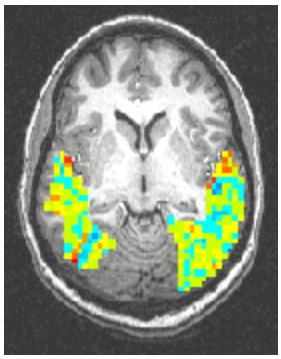


Figure 1.1.1: fMRI activations (shown with color dots) overlaid on a transverse slice of the corresponding structural MRI brain image (in grayscale). Each color dot represents a particular voxel. Top (bottom) represents the anterior (posterior) part of the brain.

or a group of neurons activate, they elicit glucose consumption and supply of oxygen-carrying blood to the area around the activations. However, the amount of the oxygen consumed is less than the amount of the oxygen supplied, leading to a change in the proportion of oxyhemoglobin and deoxyhemoglobin compared to the proportion in the resting state. This causes a change in the magnetic properties around the location of neural activations, which is then captured by the fMRI scanner as a blood-oxygenation-level-dependent (BOLD) signal. For more on the relationship between neural activities and the BOLD signal, see Logothetis et al. (2001).

The BOLD signal is temporally blurred compared to the neural activations: neural activations lasting in the order of a few hundred milliseconds can give rise to a response in the BOLD signal in the order of a few (10-15) seconds. On the other hand, a relatively high spatial accuracy can be obtained in fMRI. Current state of the art fMRI scanners can capture data with a spatial resolution of $3 \times 3 \times 3\ \mathrm{mm}^3$ for a volume element (called *voxel*), containing a population of several thousands of neurons. The resulting data are in the form of 3-dimensional images of brain activations; in figure 1.1.1, we show a 2-dimensional slice of a typical fMRI image overlaid on a structural MRI brain image. The data is typically corrupted with noise from various sources. Some of this noise can be removed through some preprocessing steps, but some amount of noise will remain even in the preprocessed data.

Just like there are variations in weights and heights across the population, there are variations caused by the differences in brain size and structure across different individuals. This gives rise to different feature spaces for different human subjects. Methods are available to map fMRI data from different brains into a common brain template. However, these methods typically introduce distortions in the data, caused by the necessary inter/extrapolations from the original voxels to the voxels in the common brain template. Furthermore, the BOLD signal also depends highly on the density of the blood vessels, and there might be differences in the vascular density at a specific location in different brains. Lastly, even though, there are common functional localities across different brains, we also need to consider that different people encounter different life experiences. It is not yet known how these different experiences reflect in the patterns of activations for a specific cognitive phenomenon, for instance, how the patterns of activations representing the semantic category food differ in native English speakers vs English-as-second-language speakers. These are some of the main challenges that need to be addressed in order to be able to effectively extract information from fMRI data across subjects in various settings.

1.2 Related Work

This section reviews works related to the main thrust of the thesis. I break down these works into works in predictive modeling of fMRI data, approaches for group analysis in fMRI, and approaches for multitask

learning or inductive transfer.

1.2.1 Predictive Modeling of fMRI Data

There have been quite a few publications regarding the application of predictive modeling for the analysis of fMRI data. I will not exhaustively cover every work that has been published and will instead focus on those works that have dealt with combining data across subjects in some sense. Most of these are in the context of classifying mental states using fMRI data, i.e. the problem of finding a function from the fMRI data to a set of mental states. An approach that has been proposed is to register the fMRI data in terms of *regions of interest* (ROIs), i.e. anatomically localized areas in the brain, each containing voxels that are thought to be highly similar. In particular, Wang et al. (2004) combined data for a sentence-picture verification study across subjects by first normalizing the data into ROI supervoxels, i.e. the average of the voxels in each ROI, and pooling the data for all the subjects based on these ROI supervoxels so that they can be used as training examples for a single Gaussian Naïve Bayes (GNB) classifier. In the context of classifying whether subjects were lying or not, Davatzikos et al. (2005) also pooled data from multiple subjects after normalization to a standard brain template (the Montreal Neurological Institute or MNI template), which in turn became the training examples of a support vector machine with a Gaussian kernel. The pooling was also done by Mourão-Miranda et al. (2006), which considered the effects of temporal compression and space selection on single-subject and multi-subject classification of fMRI data using the linear support vector machine (SVM), and by Shinkareva et al. (2008) in the context of the identification of the category as well as the actual object viewed by participants when objects from the categories tools and dwellings were used.

All of the works mentioned above concerns the classification of mental states given fMRI data. More recently, another paradigm associated with predictive modeling of fMRI data where the predictive task is to predict the fMRI activations associated with a particular mental state. In the context of this paradigm, Just et al. (2010) proposed using combining fMRI data across multiple subjects in terms of factors common across the different subjects, which were then used to predict the fMRI activations associated with concrete objects. These common factors were discovered using factor analysis on the data. In this thesis we also consider the idea of finding common factors for combining fMRI data across subjects. As will be seen later in the thesis, the approach differs from that described in Just et al. (2010) in that Just et al. (2010) performs implicit registration of the fMRI data to five brain lobes, a step not required in our approach. In addition, while Just et al. (2010) applied their approach to fMRI from only a single study, we also apply our approach to integrate fMRI data from multiple studies.

While there are a few publications that consider combining data across subjects, there has not been any publication regarding combining data across studies. In our group, there have been (unpublished) investigations of how well naïve pooling across studies works for fMRI datasets studying semantic categories (with different stimulus types for different studies) after normalization into the MNI template. In some cases, better-than-random classification accuracies can be obtained. This indicates that indeed sharing across studies is possible, and a principled way to do that will contribute significantly to the field. Nonetheless, these investigations involved multiple studies that are similar in nature, in this particular case the objects presented in the different studies were the same. This means that we can match a trial (corresponding to an object) in one study to another trial that corresponds to the same object in any of the other studies. In this thesis, we also investigate combining fMRI data from multiple studies along similar lines, i.e. we assume that the different studies have the same kinds of trials. In this thesis we also try to relax this assumption by having only some of the trials to be of the same kinds.

1.2.2 Group Analysis in fMRI

In conventional fMRI data analysis, analysis over multiple subjects is called group analysis. The main focus is to obtain reliable population inference of a particular effect, i.e. whether the effect exists across all the subjects, by accounting for variations of that effect across subjects; for instance, one might want to find out whether a particular location in the brain is significantly activated (effect in this case is activation) regardless

of the subjects when these subjects perform a certain cognitive task. The analysis is typically done using *mixed-effects models*, first suggested by Woods (1996) (an overview can be found in Penny et al. (2003)). A mixed-effect model consists of some *fixed effects* and some *random effects*. More formally, a simple version of the model assumes that for a particular location in a particular subject s, the effect β_s for that subject can be decomposed as

$$\beta_s = \beta^{(\text{fixed})} + \beta_s^{(\text{random})}, \tag{1.1}$$

The fixed effect $\beta^{(\text{fixed})}$ at the same location is shared by all the subjects in the population, while the location's random effect $\beta_s^{(\text{random})}$ is specific to each subject and represents how that subject's effect deviates from the population effect. The random effects are usually modeled as Gaussian with mean zero and variance components that need to be estimated. As will be seen in chapter 2, the parameters of the mixed-effects model can be estimated using maximum likelihood or variations of it.

Friston et al. (2002a) proposed a Bayesian formulation of the mixed-effects model for group analysis of fMRI data. In particular, they cast the problem as a particular hierarchical Bayes model and used the parametric empirical Bayes method proposed in Friston et al. (2002b) to obtain fixed-effects and random-effects estimates. The use of the Bayesian procedure was tied in with the desire to obtain maps of posterior probabilities of activations. Lazar et al. (2002) surveyed other methods from the field of meta analysis—the field concerned with combining information from a group of studies—that can be applicable in the context of fMRI data analysis to pool information over multiple subjects.

In general, as alluded to above, the main focus of group analysis of fMRIdata is to detect the significant presence of a particular effect in the population of subjects; in particular, the significance of the effect is quantified using the framework of hypothesis testing. Hence, the objective of group analysis is inherently different from the objective of predictive analysis underlying the proposed thesis. Despite this difference, ideas used in the above works can be incorporated into some of the methods that I propose to investigate, especially those involving modeling across subjects. In particular, in chapter 2, we present applications of the mixed-effects model in the predictive setting.

1.2.3 Multitask Learning/Inductive Transfer

As mentioned earlier, the problem we consider in this thesis can be framed as an instance of a general machine learning problem of learning from multiple related tasks, known as *multitask learning* (Caruana (1997)), *inductive transfer*, *transfer learning*, or *domain adaptation* (Daumé III and Marcu (2006)). The idea is by having a method that can learn to do multiple tasks is able to leverage the related information that exists across the different tasks such that its performance is better in all the tasks compared to methods that are specific to specific tasks.

There have been a few methods proposed to do multitask learning. Bakker and Heskes (2003) discussed probabilistic ways to differentiate task similarities in the context of multitask learning using neural networks. Yu et al. (2005) proposed a way to learn Gaussian processes from multiple related tasks by specifying a common multivariate Gaussian prior instantiations of related tasks taking the form of functions. Zhang et al. (2006) used a probabilistic model based on independent components analysis (Hyvärinen et al. (2001)) to model interactions between tasks. Rosenstein et al. (2005) extended the naïve Bayes model for multinomial data using a hierarchical Bayes model and apply it to meeting acceptance data. Marx et al. (2005) extended the logistic regression model for predicting a new task by using a Gaussian prior on the logistic regression coefficients across tasks and learning the parameters for this prior for the new task by using maximum likelihood over the coefficients for the related tasks. Xue et al. (2007) considered connecting logistic regression coefficients across related tasks, and enabling task clustering using the Dirichlet process mixture model. The Dirichlet process mixture model was also used by Roy and Kaelbling (2007) to enable clustering across tasks in a naïve Bayes model. A direct analogue in the Bayesian statistics field for classification across multiple tasks is the hierarchical logistic regression (see for instance chapter 16 of Gelman et al. (2003)), which uses hierarchical models to couple logistic regression coefficients across related groups of data.

One aspect that the above methods have in common is the assumption of a common feature space across tasks. As mentioned above, when doing predictive analysis of fMRI data, it might be desirable to deal directly with data with differing feature spaces without having to normalize them to a common space.

1.2.4 Relationship between the Thesis and the Related Work

With regards to the related work mentioned above, this thesis contributes the following:

- We present an application of the mixed effects model to extend the Gaussian Naïve Bayes classifier
- We apply the mixed effects model to the problem of predicting fMRI activations associated with a particular mental state
- With respect to Just et al. (2010), we present a way to find common factors in the fMRI data across subjects without having to perform implicit registration/normalization, and we also find factors that are common across studies
- We present a way to do multitask learning when the feature space of each task is different from the feature space of another task

1.3 fMRI Datasets

Here we describe the fMRI datasets used in this thesis.

1.3.1 Starplus Dataset

This dataset was collected to study differences in strategies used by different individuals in a sentence-picture verification task (Reichle et al. (2000)). It consisted of trials with presentations of sentences and pictures. We use this dataset to perform a classification experiment where we classify some brain activations data into either the sentence or the picture class.

1.3.1.1 Experiment Design

In this study, a trial consisted of the presentations of a picture of vertical arrangements of a star (*), a plus (+), and a dollar sign ($), along with a possible sentence description of the picture. In each trial, the subject was instructed to decide whether the sentence described the picture and report the decision with a button press. There were 80 trials in the experiment; in half of the trials, the picture was presented before the sentence, and in the other half, the sentence was presented before the picture. For each trial, the first stimulus was presented for four seconds followed by a four-second period of blank screen before the second stimulus was presented. The second stimulus was presented for four seconds or until the subject pressed a mouse button, whichever came first. A rest period of 15 seconds followed after the second stimulus disappeared until the start of the next trial.

1.3.1.2 fMRI Acquisition Parameters

fMRI data were acquired using a 3T fMRI scanner, with TR=1000ms. Only a subset of the brain, selected based on the expected activated areas, was captured.

1.3.1.3 Data Preprocessing

Time/slice correction, motion correction, filtering and detrending were applied to the data using the FIASCO (Eddy et al. (1996)) software. The data were further divided into 24 anatomically defined regions of interest (ROIs). Data from 13 subjects are available from this study.

1.3.2 Twocategories Dataset

This dataset was collected to study distinguishable patterns of brain activations associated objects of the categories *tools* and *buildings*, and it has been analyzed in Pereira et al. (2006). In this thesis, we use this data to perform a classification experiment where we classify the categories of the objects, i.e. a binary classification task with the tools and buildings classes.

1.3.2.1 Experiment Design

In this study, words from categories "tools" and "dwellings" were presented to each subject. There are 7 words used for each category, with the specific exemplars being

- **tools**: hammer, pliers, screwdriver, hatchet, saw, drill, wrench
- **buildings**: palace, mansion, castle, hut, shack, house, apartment

While a word was being shown, each subject was instructed to think about the properties of the word and decide which category the word belongs to. The experiment was divided into six epochs, where in each epoch was presented once, with the constraint that no two words from the same category were presented consecutively. A trial refers to the presentation of a particular word, so each epoch contained 14 trials. In each trial, the word was presented for 3 seconds, followed by an 7-to-8-second period of fixation before the next trial was presented.

1.3.2.2 fMRI Acquisition Parameters

fMRI data were acquired using a 3T fMRI scanner, with TR=1000ms and voxel size $3.125 \times 3.125 \times 6\mathrm{mm}^3$.

1.3.2.3 Data Preprocessing

Time/slice correction, motion correction, detrending/filtering were applied to the data using the SPM99 software. The data for each subject were then registered to the MNI space (Evans et al. (1993)), preserving the $3.125 \times 3.125 \times 6\mathrm{mm}^3$ voxel size. For each trial, we then averaged the activations from time points 5 to 8. The data were then normalized such that in each trial, the mean and variance across voxels were 0 and 1, respectively. Data from six subjects are available for this study.

1.3.3 Word-Picture (WP) Dataset

This dataset was collected to study the patterns of activations associated with concrete everyday objects from numerous categories. Here the stimuli are in the form of a picture and the word label for each of the objects. Using this dataset, we perform experiments predicting the brain activations associated with an arbitrary concrete object, following the analysis performed in Mitchell et al. (2008).

1.3.3.1 Experiment Design

In this study, sixty concrete nouns were presented to the subjects. The concrete nouns used are shown in table 1.3.1. The experiment was divided into six epochs, where each of the sixty words was presented once in each epoch. A presentation of a word is referred to as a trial. In each trial, the word was shown in the form of line drawing of and the word label for the object, and the subjects were instructed to think about the properties of the word being presented. In each trial, the stimulus was presented for 3 seconds, followed by a 7-second period of fixation before the next trial.

1.3.3.2 fMRI Acquisition Parameters

fMRI data were acquired using a 3T fMRI scanner, with TR=1000ms and voxel size $3.125 \times 3.125 \times 6\mathrm{mm}^3$.

Category	Exemplar 1	Exemplar 2	Exemplar 3	Exemplar 4	Exemplar 5
animals	bear	cat	cow	dog	horse
body parts	arm	eye	foot	hand	leg
buildings	apartment	barn	church	house	igloo
building parts	arch	chimney	closet	door	window
clothing	coat	dress	pants	shirt	skirt
furniture	bed	chair	desk	dresser	table
insects	ant	bee	beetle	butterfly	fly
kitchen utensils	bottle	cup	glass	knife	spoon
man-made objects	bell	key	refrigerator	telephone	watch
tools	chisel	hammer	pliers	saw	screwdriver
vegetables	carrot	celery	corn	lettuce	tomato
vehicles	airplane	bicycle	car	train	truck

Table 1.3.1: The words used in the WP and WO studies.

1.3.3.3 Data Preprocessing

Time/slice correction, motion correction, detrending/filtering were applied to the data using the SPM2 software. The data for each subject were then registered to the MNI space, preserving the $3.125 \times 3.125 \times 6\text{mm}^3$ voxel size. For each trial, we then averaged the activations from time points 4 to 7. Data from nine subjects are available for this study.

1.3.4 Word-Only (WO) Dataset

Like the Word-Picture dataset, this dataset was collected to study the patterns of activations associated with concrete everyday objects from numerous categories. However, unlike in the Word-Picture dataset, here the stimuli are in the form of only the word label for each of the objects. So in the Word-Only dataset, each object is not grounded with a particular visual depiction. Using this dataset, we also perform experiments predicting the brain activations associated with an arbitrary concrete object. This dataset has previously been analyzed in Just et al. (2010).

1.3.4.1 Experiment Design

In this study, sixty concrete nouns used in the WP study, shown in table 1.3.1, were presented to the subjects. As in the WP study, the experiment was divided into six epochs, where each of the sixty words was presented once in each epoch. However, in each trial, the word was shown in the form of word label only for the object, and the subjects were again instructed to think about the properties of the word being presented. In each trial, the stimulus was presented for 3 seconds, followed by a 7-second period of fixation before the next trial.

1.3.4.2 fMRI Acquisition Parameters

fMRI data were acquired using a 3T fMRI scanner, with TR=1000ms and voxel size $3.125 \times 3.125 \times 6\text{mm}^3$.

1.3.4.3 Data Preprocessing

Time/slice correction, motion correction, detrending/filtering were applied to the data using the SPM2 software. The data for each subject were then registered to the MNI space, preserving the $3.125 \times 3.125 \times 6\text{mm}^3$ voxel size. For each trial, we then averaged the activations from time points 4 to 7. Data from eleven subjects are available for this study; three of the subjects also participated in the WP study.

Chapter 2

Approaches Based on Hierarchical Linear Model

Abstract

In this chapter, we take the approach commonly used to perform group analysis of fMRI data—the hierarchical linear model—and use it to extend existing methods that have been used for predictive analysis of fMRI data. The idea of the hierarchical linear model is that the same parameter in different subjects have similar, but not necessarily the same, values. The variation of the parameter value is modeled as having a Gaussian distribution. The subject-specific parameter value in turn is estimated using a shrinkage estimator, balancing the subject-specific contribution with the common subject-independent contribution. Implicit in the hierarchical linear model is the assumption that each subject-specific model has the same kinds of parameters. We use the hierarchical linear model to extend the Gaussian Naïve Bayes classifier and the linear regression, and the resulting methods are applied to real fMRI data. In the classification experiment, we see the hierarchical Gaussian Naïve Bayes classifier being able to adapt to the number of training examples, in the sense that it is able to use the available cross-subject information when the number of training examples for a particular subject is small, and use more of the information available for that particular subject as the number of training examples increases. However, in the regression experiment, we do not see significant improvements using the hierarchical linear regression compared to when we train a separate linear regression model for each subject or when we train a linear regression model on the pooled data from all the subjects.

Hierarchical linear models (Raudenbush and Bryk (2001)—also known as mixed models (Demidenko (2004))—are commonly used to perform group analysis of fMRI data, mentioned in the previous chapter, mostly in the context of detecting significant activations across a group of subjects, for instance, Penny et al. (2003). The idea is that a parameter related to the fMRI data for a particular subject is similar to a corresponding parameter related to the fMRI data for another subject. Similar here means that these subject-specific parameters are based on some common value, but they are allowed to vary to some degree from this common value. Implicit in this idea is the fact that we can find a correspondence between a parameter in one subject to a parameter in another subject. This in turn constrains the application of the model to fMRI datasets that are in the same feature space.

Let us delve a little bit deeper on how the hierarchical linear model is currently used for group analysis of fMRI data. A general linear regression model can be written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{2.1}$$

where we want to find the relationship between the $N \times 1$ vector of responses $\mathbf{y}$ with the covariates, denoted as the $N \times K$ matrix $\mathbf{X}$. The relationship is captured by the $K \times 1$ vector $\boldsymbol{\beta}$, and $\boldsymbol{\epsilon}$ ($N \times 1$ vector) denotes the error in the responses. In the context of fMRI data analysis based on statistical parametric mapping (Kiebel and Holmes (2003)), equation (2.1) is referred to as the *general linear model* (GLM). In the GLM analysis of fMRI data, N represents the number of fMRI scans/images, and $\mathbf{y}$ represents a particular voxel's activation for all the scans. Each column of $\mathbf{X}$—commonly referred to as the *design matrix* in the GLM context—represents an explanatory variable for the voxel's activations, for instance, the expected BOLD response based on the stimulus timing or the drift in the BOLD response due to the scanner. In general, the explanatory variables contained in $\mathbf{X}$ are closely related to the settings used in the fMRI experiment. As mentioned above, the typical objective of the analysis to find voxels that exhibit significant fMRI activations corresponding to a particular condition. Typically the condition-of-interest is expressed as a contrast, for instance stimulus-vs-baseline contrast or contrast between stimulus 1 vs stimulus 2, and of particular importance is the coefficient (an element of $\boldsymbol{\beta}$) corresponding to the explanatory variable indicating the BOLD response that arises from the condition-of-interest. This coefficient indicates the effect of the condition-of-interest on the particular voxel's activations. If there is indeed a relationship between the condition-of-interest and the voxel's activations, the coefficient will not be zero, and to test for this, in the GLM analysis of fMRI data, the *t* statistic for this coefficient is computed.

The above discussion concerns analysis of fMRI data from a single subject. When there are multiple subjects, there is a linear regression model associated with each subject s:

$$\mathbf{y}^{(s)} = \mathbf{X}\boldsymbol{\beta}^{(s)} + \boldsymbol{\epsilon}^{(s)}. \tag{2.2}$$

where $\mathbf{y}^{(s)}$ denotes the particular voxel's activations in subject s. As can be seen in equation (2.2), the same design matrix $\mathbf{X}$ is shared by all the subjects because in a typical fMRI study the same experimental settings (stimulus timing, etc) are used for all the subjects. In order to see how the effect from the condition-of-interest manifests itself in the population of subjects, it is useful to relate the $\boldsymbol{\beta}^{(s)}$'s for all the subjects. In particular, for group analysis using the GLM, the relationship is assumed to be

$$\boldsymbol{\beta}^{(s)} = \boldsymbol{\gamma} + \boldsymbol{\eta}^{(s)}, \tag{2.3}$$

where we assume that the subject-specific coefficients $\beta^{(s)}$ arise from some population-wide coefficients $\boldsymbol{\gamma}$. This is a special case of the hierarchical linear model. In order to make inference on the effect of the condition-of-interest within the population, we look at the corresponding coefficient in $\boldsymbol{\gamma}$.

The novel contribution of the thesis described in this chapter is to adapt and apply the hierarchical linear model to other kinds of fMRI analysis that can be framed as a linear regression problem as in equation (2.1). In particular, in this chapter, we describe two scenarios. The main difference between the two scenarios and the GLM analysis is what we use as covariates (elements of $\mathbf{X}$). In the first scenario, by noting that the Gaussian Naïve Bayes classifier can be framed as a linear regression model shown in equation (2.1) with a particular set of covariates $\mathbf{X}$, we describe an extension of the classifier using the hierarchical linear model that can be applied to multiple-subject fMRI data scenario. In the second scenario, we consider the

hierarchical linear model in a context where, instead of experimental settings, the matrix of covariates $\mathbf{X}$ contains semantic information in a model that assumes that the fMRI activations reflect the meaning of concrete objects.

Here is an outline of the rest of the chapter. In section 2.1, we give a general formulation of hierarchical linear models. This formulation is then used to extend the Gaussian Naïve Bayes classifier, described in section 2.2, and extend the multivariate linear regression model, described in section 2.3. We also include in these two sections results of applying the extended methods to actual fMRI data. In section 2.4, we conclude with a discussion about the advantages and drawbacks of the approach.

2.1 Hierarchical Linear Model

The formulation in this section is based on materials in Raudenbush and Bryk (2001) and Demidenko (2004).

2.1.1 Model

Let us consider the linear regression model, where we have data from M groups (as seen above, in the fMRI context, a group typically refers to a particular human subject):

$$\mathbf{y}^{(m)} = \mathbf{X}^{(m)}\boldsymbol{\beta}^{(m)} + \boldsymbol{\epsilon}^{(m)}, \quad 1 \leq m \leq M. \tag{2.4}$$

Here the vector $\mathbf{y}^{(m)}$ is an $n^{(m)} \times 1$ vector containing the $n^{(m)}$ instances or data points in group m, $\mathbf{X}^{(m)}$ is the $n^{(m)} \times K$ matrix of covariates or predictors for group m, $\boldsymbol{\beta}^{(m)}$ is the $K \times 1$ vector of linear regression coefficients for group m, and $\boldsymbol{\epsilon}^{(m)}$ is group m's $n^{(m)} \times 1$ vector of errors. We assume

$$\boldsymbol{\epsilon}^{(m)} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_{n^{(m)}}), \tag{2.5}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix. Note that the disparate groups can have different numbers of instances, but there are the same number of covariates K for all the groups, and we also assume that variance σ^2 for all the groups is the same.

As formulated here so far, we have M distinct linear regression models, one for each group. However, if we can assume that the M groups are related, it might be desirable to link these models together. In particular, in a hierarchical linear model or a mixed model, we link the different $\boldsymbol{\beta}^{(m)}$'s as follows:

$$\boldsymbol{\beta}^{(m)} = \mathbf{W}^{(m)}\boldsymbol{\gamma} + \mathbf{u}^{(m)}, \quad 1 \leq m \leq M. \tag{2.6}$$

Here, we assume that for each group m, the associated $\boldsymbol{\beta}^{(m)}$ comes from another linear regression with group-specific covariates $\mathbf{W}^{(m)}$ ($K \times L$ matrix) and group-independent regression coefficients $\boldsymbol{\gamma}$ ($L \times 1$ vector). We assume

$$\mathbf{u}^{(m)} \sim \mathcal{N}(\mathbf{0}, \mathbf{T}), \tag{2.7}$$

for some group-independent $K \times K$ covariance matrix $\mathbf{T}$.

The model formulated above is commonly referred to as a two-level hierarchical linear model, where equations (2.4) and (2.6) are levels 1 and 2, respectively, of the model. Using the terminology used in Raudenbush and Bryk (2001), $\boldsymbol{\gamma}$ is commonly referred to as the vector of *fixed effects*, while $\mathbf{u}^{(m)}$ is commonly referred to as the vector of (level-2) *random effects*.

2.1.2 Estimation

Now we turn to the problem of estimating the parameters of a hierarchical linear model. We take a maximum-likelihood approach to obtain the parameter estimates. The parameters that we need to estimate are $\boldsymbol{\gamma}, \sigma^2$, and $\mathbf{T}$. As shown below, we can view this as a missing data problem, where we have missing data $\mathbf{u}^{(m)}, 1 \leq m \leq M$.

Substituting the right-hand side of equation (2.6) into equation (2.4), we have for each m,

$$\mathbf{y}^{(m)} = \mathbf{X}^{(m)}\mathbf{W}^{(m)}\boldsymbol{\gamma} + \mathbf{X}^{(m)}\mathbf{u}^{(m)} + \boldsymbol{\epsilon}^{(m)}, \tag{2.8}$$

which can also be written as

$$\mathbf{y}^{(m)} - \mathbf{X}^{(m)}\mathbf{u}^{(m)} = \mathbf{X}^{(m)}\mathbf{W}^{(m)}\boldsymbol{\gamma} + \boldsymbol{\epsilon}^{(m)}. \tag{2.9}$$

If we were to have the missing data $\mathbf{u}^{(m)}$, then the left-hand side of equation (2.9) would be completely observed and equation (2.9) would just be a linear regression with covariates $\mathbf{X}^{(m)}\mathbf{W}^{(m)}$ and response $\mathbf{y}^{(m)} - \mathbf{X}^{(m)}\mathbf{u}^{(m)}$. Hence, the maximum likelihood estimate for $\boldsymbol{\gamma}$ would be given by the ordinary least squares (OLS) estimate

$$\hat{\boldsymbol{\gamma}} = \left(\sum_{m=1}^{M}(\mathbf{X}^{(m)}\mathbf{W}^{(m)})^T\mathbf{X}^{(m)}\mathbf{W}^{(m)}\right)^{-1}\sum_{m=1}^{M}(\mathbf{X}^{(m)}\mathbf{W}^{(m)})^T(\mathbf{y}^{(m)} - \mathbf{X}^{(m)}\mathbf{u}^{(m)}). \tag{2.10}$$

We could then use the residuals

$$\hat{\boldsymbol{\epsilon}}^{(m)} = \mathbf{y}^{(m)} - \mathbf{X}^{(m)}\mathbf{u}^{(m)} - \mathbf{X}^{(m)}\mathbf{W}^{(m)}\hat{\boldsymbol{\gamma}} \tag{2.11}$$

to obtain the maximum-likelihood estimate for σ^2:

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{m=1}^{M}(\hat{\boldsymbol{\epsilon}}^{(m)})^T\hat{\boldsymbol{\epsilon}^{(m)}}, \tag{2.12}$$

where $N = \sum_{m=1}^{M} n^{(m)}$. The obtain the estimate for $\mathbf{T}$, we would just use $\mathbf{u}^{(m)}$:

$$\hat{\mathbf{T}} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{u}^{(m)}(\mathbf{u}^{(m)})^T. \tag{2.13}$$

Of course, in reality, we do not observe $\mathbf{u}^{(m)}$, so we resort to the expectation-maximization (EM) algorithm (Dempster et al. (1977)), which provides a way to obtain maximum-likelihood estimates in the presence of missing data. In particular, the above maximum-likelihood estimates (equations (2.10), (2.12), and (2.13)) constitute the M-step in the EM algorithm when for each term $f(\mathbf{u}^{(m)})$ dependent on $\mathbf{u}^{(m)}$, we replace it with the associated conditional expectation $\mathbb{E}[f(\mathbf{u}^{(m)})|\mathbf{y}^{(m)}, \boldsymbol{\gamma}, \sigma^2, \mathbf{T}]$. In particular, in equations (2.10), (2.12), and (2.13), the terms dependent on $\mathbf{u}^{(m)}$ (called the complete-data sufficient statistics or CDSS in Raudenbush and Bryk (2001)) are

- $(\mathbf{X}^{(m)}\mathbf{W}^{(m)})^T\mathbf{X}^{(m)}\mathbf{u}^{(m)}$
- $\mathbf{u}^{(m)}(\mathbf{u}^{(m)})^T$
- $(\mathbf{y}^{(m)})^T\mathbf{X}^{(m)}\mathbf{u}^{(m)}$
- $(\mathbf{u}^{(m)})^T(\mathbf{X}^{(m)})^T\mathbf{X}^{(m)}\mathbf{u}^{(m)}$

The conditional distribution of $\mathbf{u}^{(m)}$ is given by (see Raudenbush and Bryk (2001) for proof)

$$\mathbf{u}^{(m)}|\mathbf{y}^{(m)}, \boldsymbol{\gamma}, \sigma^2, \mathbf{T} \sim \mathcal{N}(\hat{\mathbf{u}}^{(m)}, \sigma^2(\mathbf{C}^{(m)})^{-1}), \tag{2.14}$$

where

$$\hat{\mathbf{u}}^{(m)} = (\mathbf{C}^{(m)})^{-1}(\mathbf{X}^{(m)})^T(\mathbf{y}^{(m)} - \mathbf{X}^{(m)}\mathbf{W}^{(m)}\gamma) \tag{2.15}$$

$$\mathbf{C}^{(m)} = (\mathbf{X}^{(m)})^T\mathbf{X}^{(m)} + \sigma^2\mathbf{T}^{-1}. \tag{2.16}$$

So now we can calculate the conditional expectations for our complete-data sufficient statistics:

$$\mathbb{E}\left[(\mathbf{X}^{(m)}\mathbf{W}^{(m)})^T\mathbf{X}^{(m)}\mathbf{u}^{(m)}|\mathbf{y}^{(m)},\boldsymbol{\gamma},\sigma^2,\mathbf{T}\right] = (\mathbf{X}^{(m)}\mathbf{W}^{(m)})^T\mathbf{X}^{(m)}\hat{\mathbf{u}}^{(m)} \quad (2.17)$$

$$\mathbb{E}\left[\hat{\mathbf{u}}^{(m)}(\hat{\mathbf{u}}^{(m)})^T|\mathbf{y}^{(m)},\boldsymbol{\gamma},\sigma^2,\mathbf{T}\right] = \mathbf{u}^{(m)}(\mathbf{u}^{(m)})^T + \sigma^2(\mathbf{C}^{(m)})^{-1} \quad (2.18)$$

$$\mathbb{E}\left[(\boldsymbol{\epsilon}^{(m)})^T\boldsymbol{\epsilon}^{(m)}|\mathbf{y}^{(m)},\boldsymbol{\gamma},\sigma^2,\mathbf{T}\right] = (\check{\boldsymbol{\epsilon}}^{(m)})^T\check{\boldsymbol{\epsilon}}^{(m)} + \sigma^2(\mathbf{C}^{(m)})^{-1}(\mathbf{X}^{(m)})^T\mathbf{X}^{(m)}, \quad (2.19)$$

where $\check{\boldsymbol{\epsilon}}^{(m)} = \mathbf{y}^{(m)} - \mathbf{X}^{(m)}\mathbf{W}^{(m)}\boldsymbol{\gamma} - \mathbf{X}^{(m)}\hat{\mathbf{u}}^{(m)}$. This constitutes the E-step of the EM algorithm, where for the parameters $\gamma, \sigma^2, \mathbf{T}$ we use the estimates obtained from the immediately preceding M-step.

2.1.2.1 Closed-Form Estimates in the Balanced-Design Case

The EM algorithm is an iterative procedure that needs to be run until convergence in order to obtain the estimates. However, there is a special case of the hierarchical linear model where we can obtain closed-form solutions for the maximum-likelihood estimators. This is the case where $n^{(m)} = n$ and $\mathbf{X}^{(m)} = \mathbf{X}$ for all m. In other words, in this case all the groups have the same number of instances and the same level-1 matrix of covariates, and this case is commonly referred to as the case where we have *balanced design*. Equation (2.8) becomes

$$\mathbf{y}^{(m)} = \mathbf{X}\mathbf{W}^{(m)}\boldsymbol{\gamma} + \mathbf{X}\mathbf{u}^{(m)} + \boldsymbol{\epsilon}^{(m)}. \quad (2.20)$$

In order to obtain the closed-form solutions, we start with the log-likelihood of the data (letting $\mathbf{T}^* = \frac{1}{\sigma^2}\mathbf{T}$ and omitting the constant term):

$$\ell = -\frac{1}{2}\left\{Mn\log\sigma^2 + \sum_{m=1}^{M}\left(\log|\mathbf{I}_n + \mathbf{X}\mathbf{T}^*\mathbf{X}^T| + \frac{1}{\sigma^2}(\mathbf{y}^{(m)} - \mathbf{X}\mathbf{W}^{(m)}\boldsymbol{\gamma})^T(\mathbf{I}_n + \mathbf{X}\mathbf{T}^*\mathbf{X}^T)^{-1}(\mathbf{y}^{(m)} - \mathbf{X}\mathbf{W}^{(m)}\boldsymbol{\gamma})\right)\right\}. \quad (2.21)$$

The estimate $\hat{\boldsymbol{\gamma}}$ that maximizes the log-likelihood is given by the OLS estimate (see Demidenko (2004) for proof)

$$\hat{\boldsymbol{\gamma}} = \left(\sum_{m=1}^{M}(\mathbf{X}\mathbf{W}^{(m)})^T\mathbf{X}\mathbf{W}^{(m)}\right)^{-1}\sum_{m=1}^{M}(\mathbf{X}\mathbf{W}^{(m)})^T\mathbf{y}^{(m)}. \quad (2.22)$$

In addition, the maximum-likelihood estimates for the variance parameters σ^2 and $\mathbf{T}^*$ are given by (see again Demidenko (2004) for proof)

$$\hat{\sigma}^2 = \frac{1}{M(n-K)}\sum_{m=1}^{M}(\mathbf{y}^{(m)})^T(\mathbf{I}_n - \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T)\mathbf{y}^{(m)} \quad (2.23)$$

$$\hat{\mathbf{T}}^* = \frac{1}{M\hat{\sigma}^2}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\hat{\mathbf{E}}\hat{\mathbf{E}}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} - (\mathbf{X}^T\mathbf{X})^{-1}, \quad (2.24)$$

where

$$\hat{\mathbf{E}}\hat{\mathbf{E}}^T = \sum_{m=1}^{M}(\mathbf{y}^{(m)} - \mathbf{X}\hat{\boldsymbol{\gamma}})(\mathbf{y}^{(m)} - \mathbf{X}\hat{\boldsymbol{\gamma}})^T. \quad (2.25)$$

The derivation of maximum-likelihood estimates for the variance terms σ^2 and $\mathbf{T}^*$ involves $\hat{\boldsymbol{\gamma}}$, which in turn is also quantity that is estimated. This potentially introduces bias in the resulting maximum-likelihood estimates $\hat{\sigma}^2$ and $\hat{\mathbf{T}}^*$. To alleviate this problem, *restricted maximum-likelihood* (ReML) estimation has been proposed (Harville (1977)).

In the ReML setting, the estimates for $\boldsymbol{\gamma}$ and σ^2 are as the same as the corresponding maximum-likelihood estimates, while the ReML estimate for $\mathbf{T}^*$ is given by

$$\hat{\mathbf{T}}^*_R = \frac{1}{(M-1)\hat{\sigma}^2}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\hat{\mathbf{E}}\hat{\mathbf{E}}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} - (\mathbf{X}^T\mathbf{X})^{-1}, \tag{2.26}$$

with $\hat{\mathbf{E}}\hat{\mathbf{E}}^T$ as previously defined.

There is a possibility that the closed-form estimators for $\mathbf{T}^*$ yields a matrix that is not necessarily positive semidefinite, violating the condition for a covariance matrix. To solve this problem, we follow the suggestion in Demidenko (2004) and form the eigenvalue decomposition of a particular estimate $\hat{\mathbf{T}}^* = \mathbf{P}\boldsymbol{\Lambda}\mathbf{P}^T$, and then we replace the $\hat{\mathbf{T}}^*$ with the following:

$$\hat{\mathbf{T}}^* = \mathbf{P}\boldsymbol{\Lambda}_+\mathbf{P}^T, \tag{2.27}$$

where $\boldsymbol{\Lambda}_+$ is obtained from $\boldsymbol{\Lambda}$ by replacing all the negative entries of $\boldsymbol{\Lambda}$ with zeros.

2.1.2.2 Obtaining Group-Specific Parameter Estimates

Given estimates of $\boldsymbol{\gamma}, \sigma^2$, and $\mathbf{T}$, a question that arises is what is the best estimate of $\boldsymbol{\beta}^{(m)}$ for a particular group m. To obtain this, let us first consider the quadratic term for the group m in the full-data log-likelihood:

$$\frac{1}{\sigma^2}(\mathbf{y}^{(m)} - \mathbf{X}^{(m)}\boldsymbol{\beta}^{(m)})^T(\mathbf{y}^{(m)} - \mathbf{X}^{(m)}\boldsymbol{\beta}^{(m)}) + (\boldsymbol{\beta}^{(m)} - \mathbf{W}^{(m)}\boldsymbol{\gamma})^T\mathbf{T}^{-1}(\boldsymbol{\beta}^{(m)} - \mathbf{W}^{(m)}\boldsymbol{\gamma}). \tag{2.28}$$

Expanding this expression and considering only the terms involving $\boldsymbol{\beta}^{(m)}$, we obtain

$$(\boldsymbol{\beta}^{(m)})^T\left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{X}^{(m)} + \mathbf{T}^{-1}\right)\boldsymbol{\beta}^{(m)} - 2(\boldsymbol{\beta}^{(m)})^T\left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)} + \mathbf{T}^{-1}\mathbf{W}^{(m)}\boldsymbol{\gamma}\right). \tag{2.29}$$

Taking this expression and completing the square, we find that the conditional distribution of $\boldsymbol{\beta}^{(m)}$ given the data and the parameters is given by

$$\boldsymbol{\beta}^{(m)}|\mathbf{y}^{(m)}, \boldsymbol{\gamma}, \sigma^2, \mathbf{T} \sim \mathcal{N}(\hat{\boldsymbol{\beta}}^{(m)}, \hat{\mathbf{V}}^{(m)}), \tag{2.30}$$

where

$$\hat{\boldsymbol{\beta}}^{(m)} = \hat{\mathbf{V}}^{(m)}\left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)} + \mathbf{T}^{-1}\mathbf{W}^{(m)}\boldsymbol{\gamma}\right) \tag{2.31}$$

$$\hat{\mathbf{V}}^{(m)} = \left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{X}^{(m)} + \mathbf{T}^{-1}\right)^{-1}. \tag{2.32}$$

A way to obtain an estimate for $\boldsymbol{\beta}^{(m)}$ is by taking the mode of this conditional distribution. Since the distribution is Gaussian, the mode is equal to the mean, and the estimate obtained this way is given by $\hat{\boldsymbol{\beta}}^{(m)}$. This kind of estimate is commonly referred to as the *empirical Bayes* estimate (Morris (1983)), since in essence we use the mode of the conditional posterior using estimates of the model parameters.

Using the Sherman-Morrison-Woodbury formula (shown in this chapter's appendix), we can rewrite equation (2.31) as

$$\hat{\boldsymbol{\beta}}^{(m)} = \boldsymbol{\Lambda}^{(m)}\hat{\boldsymbol{\beta}}^{(m)}_{\text{OLS}} + (\mathbf{I}_K - \boldsymbol{\Lambda}^{(m)})\hat{\tilde{\boldsymbol{\beta}}}^{(m)}, \tag{2.33}$$

where

$$\mathbf{\Lambda}^{(m)} = \mathbf{T}(\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1} \quad (2.34)$$

$$\hat{\boldsymbol{\beta}}^{(m)}_{\mathrm{OLS}} = ((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)} \quad (2.35)$$

$$\hat{\hat{\boldsymbol{\beta}}}^{(m)} = \mathbf{W}^{(m)}\gamma. \quad (2.36)$$

So the empirical Bayes estimator of $\boldsymbol{\beta}^{(m)}$ can be seen as the weighted combination of the OLS estimator $\hat{\boldsymbol{\beta}}^{(m)}_{\mathrm{OLS}}$ and the level-2 estimator $\hat{\hat{\boldsymbol{\beta}}}^{(m)}$. This form of estimator is also commonly referred to as the *shrinkage* estimator, where the idea is that we are shrinking the level-1 estimate for each group toward the group-independent level-2 estimate.

2.2 Hierarchical Gaussian Naïve Bayes Classifier

One of the classifiers that have been successfully applied to classify fMRI data is the Gaussian Naïve Bayes (GNB) classifier Mitchell et al. (2004). The GNB classifier is a probabilistic generative classifier in the sense that it is based on the model of how data is generated under each class. In this chapter, after we review the GNB classifier, we describe its extension for multiple-subject fMRI data using the hierarchical linear model; we call the resulting classifier the hierarchical Gaussian Naïve Bayes (HGNB) classifier. We also compare the performance of the original GNB classifier with the HGNB classifier on a couple fMRI datasets.

2.2.1 Gaussian Naïve Bayes Classifier

The Bayes classifier chooses the class c among K classes $(c_k)_{1\le k\le K}$ which maximizes the posterior probability of the class given the data $\mathbf{y}$:

$$c = \arg\max_{c_k} P(C = c_k|\mathbf{y}) \propto \arg\max_{c_k} P(C = c_k)p(\mathbf{y}|C = c_k).$$

The data $\mathbf{y}$ is assumed to be a vector of length n composed of *features* $y_j, 1 \le j \le n$. The Naïve Bayes classifier makes the additional assumption that the class-conditional probability for each feature j is independent. In other words,

$$p(\mathbf{y}|C = c_k) = \prod_{j=1}^{n} p(y_j|C = c_k).$$

If, in addition to the above, we have the assumption that for each feature j,

$$p(y_j|C = c_k) = \mathcal{N}(y_j|\theta_j^{(k)}, (\sigma_j^{(k)})^2),$$

i.e. if we assume that the class-conditional density for each feature j with respect to class k is a Gaussian with mean $\theta_j^{(k)}$ and variance $(\sigma_j^{(k)})^2$, we have what is called the Gaussian Naïve Bayes (GNB) classifier (Mitchell et al. (2004)).

In a typical application of the GNB classifier, the classifier is trained by obtaining estimates $\hat{\theta}_j^{(k)}$ and $(\hat{\sigma}_j^{(k)})^2$ for each feature j from the training data. Now, I describe two maximum-likelihood methods for learning estimates for the parameters of the GNB classifier that have been previously proposed in the context of classification of fMRI data. More precisely, we use the maximum-likelihood estimates for $\theta_j^{(k)}$, while we use the unbiased estimates for $(\sigma_j^{(k)})^2$, which differ by a factor of $\frac{n_s-1}{n_s}$ ($\frac{n-1}{n}$ in the pooled method) from the maximum-likelihood estimates.

Individual Method (Mitchell et al. (2004)) This method estimates parameters separately for each human subject. That is, for each class k, feature j, and subject s,

$$\hat{\theta}_{sj}^{(k)} = \frac{1}{n_s} \sum_{i=1}^{n_s} y_{sji}^{(k)}$$
$$(\hat{\sigma}_{sj}^{(k)})^2 = \frac{1}{n_s - 1} \sum_{i=1}^{n_s} (y_{sji}^{(k)} - \hat{\theta}_{sj}^{(k)})^2.$$

Note that there is no incorporation of information from the other subjects' data. When there is only one training example, in order to avoid degenerate situations, I use 0.001 as the variance estimate. The classifier learned using this method will be referred to as the *GNB-indiv* classifier.

Pooled Method (Wang et al. (2004)) This method assumes that all the data comes from one subject, or equivalently, that there exists no variations across subjects after normalization of the data to a common template. That is, for each class k, feature j, and for all subjects,

$$\hat{\theta}_{j}^{(k)} = \frac{1}{n} \sum_{s=1}^{S} \sum_{i=1}^{n_s} y_{sji}^{(k)}$$
$$(\hat{\sigma}_{j}^{(k)})^2 = \frac{1}{n - 1} \sum_{s=1}^{S} \sum_{i=1}^{n_s} (y_{sji}^{(k)} - \hat{\theta}_{sj}^{(k)})^2,$$

where $n = \sum_{s=1}^{S} n_s$. The estimates are the same for all subjects, and inherently, the method ignores possible variations across subjects. The classifier learned using this method will be referred to as the *GNB-pooled* classifier.

2.2.2 Using the Hierarchical Linear Model to Obtain The Hierarchical Gaussian Naïve Bayes Classifier

We see that the individual method corresponds to one extreme where there is no connection between the different groups' parameters, while the pooled method corresponds to another extreme where there is only one set of parameters for all the groups. Taking this into account, we now adapt the hierarchical linear model in order to obtain parameter estimates for the GNB classifier that provide a balance between these two extremes.

We consider each subject as a group in the context of the hierarchical linear model. That means if we have S subjects, we have $M = S$ groups. To simplify the notation, we omit the variables for feature (j) and class (k), with the understanding that we are modeling parameters for a specific feature j and a specific class k. We also incorporate the convention used in section 2.1 and put the group index as a superscript inside parentheses. In other words, $\hat{\theta}_s$ in section 2.2.1 becomes $\hat{\theta}^{(s)}$, and $\hat{\sigma}_s^2$, becomes $(\hat{\sigma}^{(s)})^2$. Furthermore, we assume that the variance in each subject is the same, so for all subject s, $(\hat{\sigma}^{(s)})^2 = \hat{\sigma}^2$.

We represent the data instances for each subject s with a vector $\mathbf{y}^{(s)}$. Equations (2.4) and (2.6) then represents the model used by the Gaussian Naïve Bayes classifier, replacing m with s, and where we have

- $\boldsymbol{\beta}^{(s)}$ is a scalar, corresponding to $\theta^{(s)}$
- $\mathbf{X}^{(s)}$ is the $n_s \times 1$ vector of all 1s, also denoted by $\mathbf{1}$
- $\mathbf{W}^{(s)}$ is the scalar 1
- $\boldsymbol{\gamma}$ is a scalar parameter, replaced by μ
- $\mathbf{T}$ is a scalar variance term, replaced by τ^2 (the corresponding $\mathbf{T}^*$ is replaced by $(\tau^*)^2$)

We can use these in conjunction with any of the estimation procedures described in section 2.1 in order to obtain parameter estimates that can be used in the context of the GNB classifier. In particular, when we assume the same number of instances n for all the subjects, this is a case where we have balanced design and we can adapt the closed-form solutions given by equations (2.22), (2.23), and (2.26), using the ReML estimator for $\mathbf{T}^*$, to obtain

$$\hat{\mu} = \frac{1}{Sn}\sum_{s=1}^{S}\sum_{i=1}^{n} y_i^{(s)} \tag{2.37}$$

$$\hat{\sigma}^2 = \frac{1}{S(n-1)}\sum_{s=1}^{S}(\mathbf{y}^{(s)})^T(\mathbf{I}_n - \frac{1}{n}\mathbf{1}\,\mathbf{1}^T)\mathbf{y}^{(s)} \tag{2.38}$$

$$(\hat{\tau}^*)^2 = \frac{1}{(S-1)\hat{\sigma}^2}\frac{1}{n^2}\mathbf{1}^T\hat{\mathbf{E}}\hat{\mathbf{E}}^T\mathbf{1} - \frac{1}{n}, \tag{2.39}$$

where now $\hat{\mathbf{E}}\hat{\mathbf{E}}^T = \sum_{s=1}^{S}(\mathbf{y}^{(s)} - \mathbf{1}\hat{\mu})(\mathbf{y}^{(s)} - \mathbf{1}\hat{\mu})^T$. As in the general case, we also need to worry about $\hat{\tau}^2$ being improper, i.e. negative. If that is the case, we reset it to 0.001.

After obtaining the above parameters, we can then find the estimate for $\theta^{(s)}$ by adapting the shrinkage equation (2.33):

$$\hat{\theta}^{(s)} = \lambda^{(m)}\frac{1}{n}\mathbf{1}^T\mathbf{y}^{(s)} + (1-\lambda^{(m)})\mu, \tag{2.40}$$

where

$$\lambda^{(m)} = \frac{\tau^2}{\tau^2 + \frac{\sigma^2}{n}}. \tag{2.41}$$

This means that $\hat{\theta}^{(s)}$ is an empirical Bayes estimate of $\theta^{(s)}$. $\hat{\theta}^{(s)}$ and $\hat{\sigma}^2$ can then be used as parameters of the GNB classifier as described before. Because of the use of the hierarchical linear model, we call the classifier using the procedure described here the *hierarchical Gaussian Naïve Bayes* (HGNB) classifier.

2.2.3 Experiments

To see how the HGNB classifier performs and how it compares with the individual and the pooled methods, we perform experiments with actual fMRI data. In particular, we run experiments on two fMRI datasets—the starplus and the twocategories datasets—to address the following questions:

- Does the HGNB classifier yield better accuracies compared to the individual and pooled methods?
- How does the accuracy of the HGNB classifier vary with the number of training examples?

2.2.3.1 Starplus dataset

Overview The starplus dataset is described in section 1.3.1. In this experiment, we take fMRI images captured in the first eight seconds of each trial, and the task is to classify the type of stimulus presented (sentence or picture) based on the fMRI data from the first eight seconds. Since one fMRI image is captured every 500ms, for a particular subject we have 16 fMRI images for each trial. Furthermore, for each of the 16 fMRI images, we take the average of the activations across the voxels in the calcarine sulcus region of interest. So for each trial in a particular subject, we have 16 numbers representing that trial. In other words, each subject's data has 16 features.

Experimental setup We perform cross-validation (CV) to evaluate how well we can classify the instances or data points. For each subject, we have 40 data points, 20 corresponding to the sentence presentations and the other 20 corresponding to the picture presentations. We divide these data points into two folds, each fold containing 10 data points from the sentence class and 10 data points from the picture class. One fold is designated for training while the other fold is designated for testing. From the training fold, we vary the number of training examples available for each class from 1 to 10. For methods that integrate data across multiple subjects (pooled method and HGNB), we perform a similar procedure, i.e. we choose the same number of data points available for training from each of the other subjects' data. So there are the same number of data points from each subject's data available for training.

In this experiment, the evaluation is performed using three classifiers:

- GNB-indiv: the GNB trained using the individual method
- GNB-pooled: the GNB trained using the pooled method to incorporate data from all the subjects
- HGNB: the hierarchical GNB classifier

Results We perform the experiment outlined above for each subject. The experiment is repeated 10 times, with different assignments of data points to folds in each repetition. Figure 2.2.1 shows the accuracies, averaged across subjects and repetitions, of the three classifiers as the number of training examples per class varies, with the error bars showing the standard deviations. We can see in the figure that the GNB-pooled and HGNB classifiers are more effective compared to the GNB-indiv classifier in the regime of low numbers of training examples. However, as the number of training examples increases, the HGNB and the GNB-indiv classifiers outperform the GNB-pooled classifier. In addition, both the HGNB and the GNB-indiv classifiers reach the same level of accuracies.

This is to be expected from the form of the estimate for the parameter $\theta^{(s)}$ as shown in equation (2.40). In this equation, we can see that we are trading off between the contribution of the data coming from the main subject and the contribution of the data from the other subjects. In particular, based on equation (2.41) for $\lambda^{(m)}$, we see that the more training examples we have from the main subject, the more weight we put on the contribution from the main subject, and correspondingly the less weight we put on the contribution from the other subjects. So when there are only a few training examples available for the main subject, we expect the accuracies to be similar to those of the GNB-pooled classifier and as the number of training examples for the main subject increases, we expect the accuracies to converge to those of the GNB-indiv classifier. The expectation is confirmed in figure 2.2.1.

2.2.3.2 Twocategories dataset

Overview The twocategories dataset is described in section 1.3.2. The task is to classify the category (tools or buildings) of the stimulus being presented given the fMRI data for a particular trial. Only voxels that are present in all the subjects are considered.

Experimental setup We again perform cross-validation (CV). There are 84 data points, 42 corresponding to the tools category and the other 42 corresponding to the buildings category. As in the experiment involving the starplus dataset, we divide the data points into two folds, and in each fold, there are the same number of data points for each class. One fold is designated as the training fold and the other is designated as the test fold. From the training fold, we vary the number of training examples available for each class from 1 to 21, and in the case of methods integrating fMRI data across multiple subjects, the data from each of the other subjects also contain the same number of examples per class. We also perform voxel selection by selecting 300 voxels that has the highest Fisher median class separation score[1] (Saito and Coifman (1995)). The scoring is based on the training examples of the subject of interest only.

We perform the experiment using the three classifiers used in the starplus experiment.

[1]The Fisher median class separation score for a particular voxel v is computed as follows. Let there be C classes, and let π_c be the proportion of instances for class c. We assume that there are n_c instances for class c, and represent the i-th instance for class c as $y_i^{(c)}$. The Fisher median class separation score is the quantity:

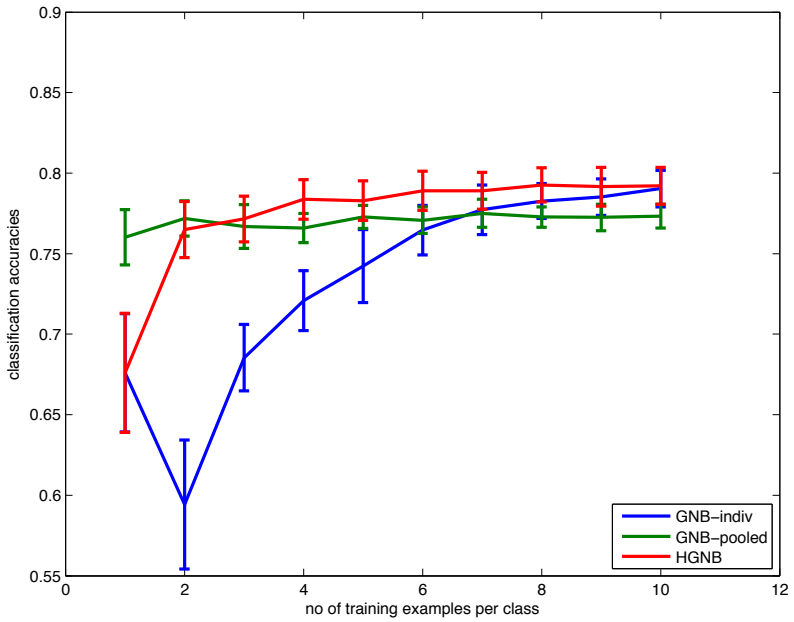


Figure 2.2.1: Accuracies of the GNB-indiv (blue), GNB-pooled (green), and HGNB (red) on the starplus datasets, averaged across 13 subjects and 10 repetitions, as a function of the number of training examples per class. The error bars indicate standard errors.

Results We perform the described experiment for each subject, repeating 10 times with the fold assignments randomized. Figure 2.2.2 show the accuracies averaged across subjects and repetitions as the number of training examples per class varies. We see trends similar to those seen in the starplus experiment. Again, in the regime of low number of training examples, GNB-pooled and HGNB outperform GNB-indiv. As the number of training examples increases, the accuracies of all three classifiers increase. However, the increase in accuracies for the GNB-pooled classifier is slow, and when more than 10 training examples per class are available, the GNB-pooled classifier underperforms both GNB-indiv and HGNB. On the other hand, the HGNB classifier matches or marginally outperforms the GNB-indiv classifier when there are a high number of training examples.

2.2.4 Discussion

Based on our experiments, we see that GNB-pooled is better than GNB-indiv when there are low number of training examples for each subject, but GNB-pooled is worse than GNB-indiv when there are more training examples available. On the other hand, the HGNB classifier achieves the best of these two extremes: its performance matches that of GNB-pooled in the case of low number of training examples, and it matches that of GNB-indiv in the case of high number of training examples. As noted when we discuss the results for the starplus dataset, we can see why by looking at the shrinkage equation (2.40). The equation says that the estimate for the parameter obtained in the HGNB classifier is a compromise between the estimate of the individual method and the estimate of the pooled method. When n is low, the estimate is weighted more heavily toward the individual estimate, while when n is high, it is weighted more heavily toward the

$$\frac{\sum_{c=1}^{C} \pi_c \left| \text{median}(y_1^{(c)}, \cdots, y_{n_c}^{(c)}) - \text{median}(\text{median}(y_1^{(1)}, \cdots, y_{n_1}^{(c)}), \cdots, \text{median}(y_1^{(C)}, \cdots, y_{n_C}^{(C)})) \right|}{\sum_{c=1}^{C} \pi_c \text{MAD}(y_1^{(c)}, \cdots, y_{n_c}^{(c)})},$$

where MAD stands for median absolute deviation.

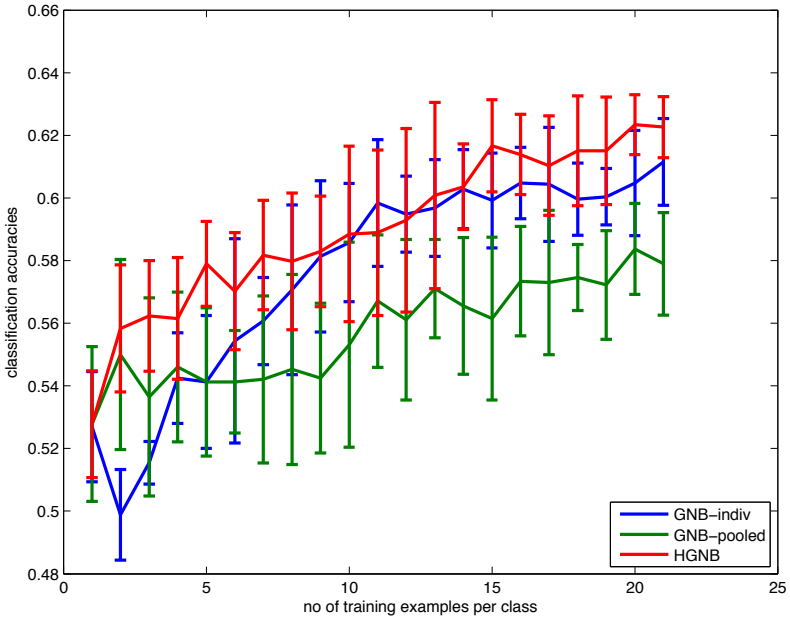


Figure 2.2.2: Accuracies of the GNB-indiv (blue), GNB-pooled (green), and HGNB (red) on the twocategories datasets, averaged across 6 subjects and 10 repetitions, as a function of the number of training examples per class. The error bars indicate standard errors.

pooled estimate. This also means that the HGNB classifier does not offer much benefit over the GNB-indiv classifier when there are a lot of training examples.

We now address the questions stated previously:

- Does the HGNB classifier yield better accuracies compared to the individual and pooled methods?

  When the number of training examples is low, the accuracies of the HGNB classifier are better than those of the GNB-indiv method and comparable to those of the GNB-pooled method. On the other hand, when there are more training examples, the accuracies of the HGNB classifier are comparable to those of the GNB-indiv method, while in this regime, the GNB-pooled is sub-optimal. In the limit of infinite number of training examples, we expect the HGNB classifier to be practically the same as the GNB-indiv classifier.

- How does the accuracy of the HGNB classifier vary with the number of training examples?

  The accuracy of the HGNB classifier increases as the number of training examples increases, until it converges to the accuracy of the GNB-indiv method.

2.3 Hierarchical Linear Regression

Next we consider the application of the hierarchical linear model in linear regression. In particular, we specialize the hierarchical linear model by letting $\mathbf{W}^{(m)}$ be the $K \times K$ identity matrix $\mathbf{I}_K$ for each m. $\boldsymbol{\gamma}$ can then be considered as the group-averaged linear regression coefficients. The estimation procedure for the parameters proceeds naturally from those described in section 2.1.

2.3.1 Experiments

Mitchell et al. (2008) propose a computational model associated with the meaning of concrete nouns, shown in figure 2.3.1. The model that the fMRI activations associated with the meaning of concrete nouns can be characterized by some predefined semantic features, denoted as circles in the figure, and which we refer to as *base features*. In particular, Mitchell et al. (2008) use as semantic features the co-occurrence counts of the concrete nouns with a set of 25 verbs as derived from some large text corpus data. The semantic features are mapped to each voxel's activations linearly, and the mappings are learned using multivariate linear regression. Mitchell et al. (2008) show that this model can be used to predict the fMRI activations associated with novel concrete nouns with better-than-random predictive accuracies.

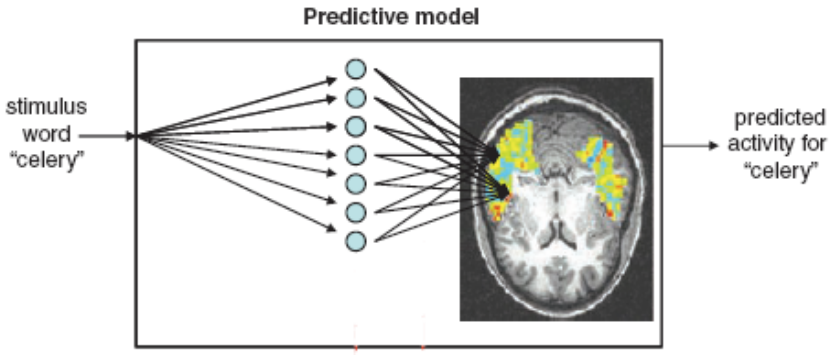


Figure 2.3.1: Predictive model of fMRI activations associated with concrete nouns proposed by Mitchell et al. (2008)

In the baseline model of Mitchell et al. (2008), there is a separate mapping from the base features to the fMRI activations of a particular subject in a particular study, and each of the mapping is learned separately without any consideration about the other mappings. In this case study, we ask the question whether we can improve the predictive accuracies by linking the regression coefficients for the different subjects together using the hierarchical linear model. We also compare it with the results when we pool all the subjects' data together. We perform the experiment on the WP dataset described in section 1.3.3.

2.3.1.1 Semantic features

The way different words are used or not used together is closely related to what each of the words mean. Using this idea, Mitchell et al. (2008) used as predefined semantic features the co-occurrence counts of the 60 stimulus words used in the WP study with a set of reference words consisting of 25 handpicked verbs. In other words, in their experiments, there are 25 features, one for each verb in the reference set, and each feature j for a stimulus word w is the number of times the word w co-occurs with the corresponding verb j. Mitchell et al. (2008) obtained these co-occurrence counts from a corpus of English web pages provided by Google, in particular by determining whether the word pair of interest occur together within five tokens of each other.

For the experiment described in this section, we also use the features used in Mitchell et al. (2008), which we refer to as the 25verb features. Because raw co-occurrence counts can take values in the order of thousands or more, as in Mitchell et al. (2008), we normalize the 25verb features so that for each instance the feature vector has length one.

2.3.1.2 Evaluation

We evaluate the predictive performance of each of the approaches by performing the cross-validation (CV) scheme described in Mitchell et al. (2008). In this scheme, for each CV fold, we hold out two words out of the sixty words used in the WP and WO datasets. In each fold, we first perform voxel selection on each subject's data. The voxel selection is performed by first calculating a stability score for each of the voxel over the 58 remaining words in the fold. The stability score indicates whether the voxel's activations for the 58 remaining words are roughly consistent across the six runs/presentations in the corresponding fMRI experiments, and it is computed by looking at the average of the pairwise correlation of the activations in each possible pair of runs. For each subject's data, we find 120 of the most stable voxels based on the computed stability scores, and then we find the union of each subject's 120 most stable voxels. Note that the most stable voxels for one subject's data do not necessarily overlap with those for another subject's data, and the number of voxels might differ from one fold to the next.

After selecting the voxels used for this fold, we train a model using the fMRI data along with the predefined semantic features. The trained model is then used to generate the predicted fMRI activations for the two held-out words. We compare the predicted activations with the true observed fMRI activations using the cosine similarity metric. In particular, in each fold we have the (held-out) true activations for word 1 and word 2 (denoted by true(word1) and true(word2)), and the model generates predicted activations for these words (denoted by pred(word1) and pred(word2)). We then calculate four numbers

- dist(true(word1), pred(word1))
- dist(true(word2), pred(word2))
- dist(true(word1), pred(word2))
- dist(true(word2), pred(word1))

where dist(v1, v2) is the cosine of the angle between vectors v1 and v2. If

$$\begin{aligned}&\text{dist(true(word1), pred(word1))} + \text{dist(true(word2), pred(word2))} > \\ &\quad \text{dist(true(word1), pred(word2))} + \text{dist(true(word2), pred(word1))}\end{aligned}$$

we assign the score 1 to the fold, indicating that the correct prediction has been made. Otherwise, the score 0 is assigned, indicating a wrong prediction. In essence, this measures whether the model predicts the fMRI images for the two held-out words well enough to distinguish which held-out word is associated with which of the two held-out images. We can then aggregate the scores in all the folds and compute the percentage of score 1, which serves to indicate how accurate the model is.

Given an accuracy estimate obtained by the procedure outlined above, two questions arise:

1. Does the accuracy estimate indicate that the model can make significantly better predictions compared to purely random predictions?
2. What is the uncertainty associated with the accuracy estimate?

The first question is addressed in Mitchell et al. (2008) using an approach similar to the permutation test, and they find that accuracies above 0.62 are statistically significant at $p < 0.05$. A procedure to address the second question is described later in section 4.2.3, in particular, by trying to find a confidence interval on the accuracy estimate. Because of the computationally intensive nature of the confidence interval estimation procedure, we do not perform this procedure for the results described in this section.

In this section, we try these four methods:

- LR-indiv: the baseline method of Mitchell et al. (2008) applied individually for each subject

- LR-pooled: the baseline method of Mitchell et al. (2008) applied to the pooled data from all the subjects
- HLR: the hierarchical linear regression method
- HLR-diag: the hierarchical linear regression method with $\mathbf{T}$ constrained to be a diagonal matrix

In the experiments considered here, all the subjects share the same design matrix $\mathbf{X}$, so this is a case where we have balanced design. So we obtain the parameters for the HLR and the HLR-diag methods using the closed-form for the balanced-design case as described in section 2.1. In addition, to obtain the diagonal-matrix estimate for $\mathbf{T}$ for the HLR-diag method, we take the estimate obtained using the procedure described in section 2.1 and set the off-diagonal terms to zero.

2.3.2 Results

We obtain the following accuracy estimates for the four methods:

- LR-indiv: 0.7917
- LR-pooled: 0.8060
- HLR: 0.7823
- HLR-diag: 0.7922

Each accuracy estimate is the average of the accuracy estimates across the 9 subjects in the WP dataset.

We see marginal differences in accuracies across the four methods. In particular, we some marginal improvement in accuracy when we pool the data across all the subjects. On the other hand, the accuracy using the HLR method is worse compared to both the LR-indiv and the LR-pooled methods, while the accuracy of the HLR-diag method is comparable to the accuracy of the LR-indiv method.

Here we do not vary the number of training examples used, so we cannot see the evolution of the accuracies with different numbers of training examples. Nonetheless, the results are somewhat striking. The fact that the LR-pooled method yields better accuracy compared to the LR-indiv method might suggest that the accuracies of the HLR and the HLR-diag methods should also be better compared to that of the LR-indiv, but we do not see that in the actual results. In the next section, we discuss possible explanations for what we see in the results.

2.3.3 Discussion

We do not see much difference in terms of the HLR methods compared to the LR-indiv and LR-pooled methods. In fact, we do get marginally worse accuracy when using the HLR method, and when the HLR-diag method is used instead, there is a marginal improvement in accuracy. This suggests that the hierarchical linear model as described here might not be sufficient as a means to share information across subjects, and we might need to add more constraints or stronger priors to some of the parameters in order to make the sharing of information more effective in the setting used here. In particular, in light of the accuracies for both the HLR and the HLR-diag methods, it might be the case that when the covariance matrix $\mathbf{T}$ is allowed to be general (the HLR case), there are not enough instances in the data to make the parameter estimates to be reliable. The fact that we can obtain a marginal improvement when using the HLR-diag method (compared to when using the HLR method) suggests that there is value in constraining what $\mathbf{T}$ should be. Furthermore, the constraint that $\mathbf{T}$ is a diagonal matrix might not be the most appropriate, and there might be other kinds of constraints that will lead to even better accuracies. For instance, when working with fMRI data, another constraint that can be used is to assume that parameters corresponding to nearby voxels should have similar values. In general, prior knowledge of the domain should guide the choice of constraint for $\mathbf{T}$, and also potentially for other parameters.

2.4 Summary

Let us now summarize the results of applying the hierarchical linear model in the context of predictive fMRI analysis across multiple subjects. We see that when used to extend the Gaussian Naïve Bayes classifier, the hierarchical linear model enables us get the best of both the individual and the pooled methods. However, the benefits are not so clear when we use the hierarchical linear model in the context of linear regression, at least for the experiments considered in this chapter. We also see that in the limit of infinite training examples, the HGNB classifier converges to the GNB classifier. However, whether in the general hierarchical linear model the empirical Bayes estimate converges to the OLS estimate is not apparent in equation (2.33), since the shrinkage factor $\mathbf{\Lambda}^{(m)}$ does not appear to converge to the identity matrix in the general case as the number of training examples increases. Nonetheless, based on the consistency of both the empirical Bayes estimate (this follows from the consistency of the posterior distribution, see chapter 4 and appendix B in Gelman et al. (2003), for instance) and the OLS estimate (this follows from the consistency of maximum-likelihood estimates, of which the OLS estimate is a case), as the number of training examples increases, we expect these two kinds of estimate to converge. Although we do not prove this formally, this means that as the number of training examples for group m increases, we expect the expression $((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1}$ to converge to the zero matrix.

One aspect of the hierarchical linear model that we have not touched on is the dependence on the number of groups, or the number of subjects in the fMRI context. The level-2 variance parameter $\mathbf{T}$ indicates how the group-specific parameters vary off of the common level-2 mean parameter $\mathbf{W}^{(m)}\boldsymbol{\gamma}$. When the number of groups is low, the estimate of $\mathbf{T}$ might be unstable, leading to a poor quality of the estimates of all parameters in the model. One guideline based on Gelman (2006) is to ensure that the number of groups is at least 5. Gelman (2006) also describe ways to handle the situation when the number of groups is low.

One restriction when trying to use the hierarchical linear model is to ensure that all the groups have the same kinds of features. However, especially in the context of predictive analysis of fMRI data, it might be desirable to have methods that do not have this restriction. The reasons include the fact that this usually necessitates the registration of all the subjects' brain into a common brain, which might introduce distortions in the process, and also that even after anatomical registration there might still be variations in terms of the actual activations across the different subjects, which are not accounted for fully by the hierarchical linear model. In the next chapter we consider a way to do this.

2.A Derivation of equation (2.33) from equation (2.31)

Let us restate equation (2.31) with $\hat{\mathbf{V}}^{(m)}$ expanded:

$$\hat{\boldsymbol{\beta}}^{(m)} = \left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{X}^{(m)} + \mathbf{T}^{-1}\right)^{-1}\left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)} + \mathbf{T}^{-1}\mathbf{W}^{(m)}\boldsymbol{\gamma}\right). \tag{2.42}$$

We use the Sherman-Morrison-Woodbury formula to find another expression for $\left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{X}^{(m)} + \mathbf{T}^{-1}\right)^{-1}$. One variation of the Sherman-Morrison-Woodbury formula states that

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1}, \tag{2.43}$$

where $\mathbf{I}$ is the identity matrix. We use this formula with $\mathbf{A} = \mathbf{T}^{-1}, \mathbf{U} = \frac{1}{\sigma^2}((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)}), \mathbf{V}^T = \mathbf{I}$ to obtain

$$\left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{X}^{(m)} + \mathbf{T}^{-1}\right)^{-1} = \mathbf{T} - \mathbf{T}\frac{1}{\sigma^2}((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})(\mathbf{I} + \mathbf{T}\frac{1}{\sigma^2}((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)}))^{-1}\mathbf{T}. \tag{2.44}$$

Also, using the property of matrix inverse, the inverse term on the right-hand side of equation (2.44) can be written as

$$\begin{aligned}\left(\mathbf{I} + \mathbf{T}\frac{1}{\sigma^2}((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})\right)^{-1} &= \left((\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1} + \mathbf{T})\frac{1}{\sigma^2}((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})\right)^{-1} &(2.45)\\ &= \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1}(\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}. &(2.46)\end{aligned}$$

This means we can rewrite the right-hand side of (2.44) as (removing terms that cancel each other)

$$\mathbf{T} - \mathbf{T}(\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\mathbf{T}. \tag{2.47}$$

Noting that

$$\begin{aligned}\mathbf{I} - (\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\mathbf{T} &= (\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}(\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1}) &(2.48)\\ &\quad -(\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\mathbf{T} &(2.49)\\ &= (\mathbf{T} + \sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1}, &(2.50)\end{aligned}$$

equation (2.42) becomes

$$
\begin{aligned}
\hat{\boldsymbol{\beta}}^{(m)} &= (\mathbf{T}-\mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\mathbf{T})\left(\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)}+\mathbf{T}^{-1}\mathbf{W}^{(m)}\boldsymbol{\gamma}\right) && (2.51)\\
&= (\mathbf{T}-\mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\mathbf{T})\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)} && (2.52)\\
&\quad +(\mathbf{T}-\mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\mathbf{T})\mathbf{T}^{-1}\mathbf{W}^{(m)}\boldsymbol{\gamma} && (2.53)\\
&= \mathbf{T}(\mathbf{I}-(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\mathbf{T})\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)} && (2.54)\\
&\quad +(\mathbf{I}-\mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1})\mathbf{W}^{(m)}\boldsymbol{\gamma} && (2.55)\\
&= \mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1}\frac{1}{\sigma^2}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)} && (2.56)\\
&\quad +(\mathbf{I}-\mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1})\mathbf{W}^{(m)}\boldsymbol{\gamma} && (2.57)\\
&= \underbrace{\mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}}_{\boldsymbol{\Lambda}^{(m)}}\underbrace{((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1}(\mathbf{X}^{(m)})^T\mathbf{y}^{(m)}}_{\hat{\boldsymbol{\beta}}^{(m)}_{\mathrm{OLS}}} && (2.58)\\
&\quad +(\mathbf{I}-\underbrace{\mathbf{T}(\mathbf{T}+\sigma^2((\mathbf{X}^{(m)})^T\mathbf{X}^{(m)})^{-1})^{-1}}_{\boldsymbol{\Lambda}^{(m)}})\underbrace{\mathbf{W}^{(m)}\boldsymbol{\gamma}}_{\hat{\hat{\boldsymbol{\beta}}}^{(m)}}, && (2.59)
\end{aligned}
$$

so we obtain equation (2.33).

Chapter 3

Approaches Based on Factor Analysis

Abstract

Approaches based on the hierarchical linear model to jointly analyze fMRI data from multiple subjects and studies require that the data are in a common feature space. Furthermore, the model parameter estimates for those approaches converge to the estimates when the individual datasets are analyzed separately as the number of training instances increases. We now present several approaches based on linear factor analysis that do not require that the multiple datasets be in a common feature space. We also present several ways to obtain parameter estimates in this approach, with an emphasis on principal component and canonical correlation analyses. To handle non-matching instances, an approach based on nearest neighbors is described. The next chapter shows the efficacy of this approach on real fMRI data.

We have seen ways to analyze fMRI datasets jointly across multiple subjects using the hierarchical linear model. A major constraint of those approaches is that all the datasets need to be in the same feature space. So normalization to a common brain is a requirement for hierarchical Bayes approaches. We also see that in the limit of infinite training instances, the estimates given by the presented hierarchical Bayes approaches are the same as estimates given by their individual counterparts. So the biggest advantage provided by hierarchical Bayes approaches is that they allow us to leverage data from other datasets when the number of instances available for a particular dataset is low. Given these findings, we ask the questions:

- Can we integrate multiple fMRI datasets without the need for normalization to a common feature space?

  This question is pertinent especially for fMRI because there are anatomical and functional variations existing in different subjects' brains. We can normalize the different subjects' brains to a common brain using image registration procedures, but they are not the optimal solution because they are hardly perfect and might introduce additional distortions in the data, plus these registration procedures do not account for functional variations that might exist in different subjects' brains.

- Is there a method for integrating multiple fMRI datasets that provides an advantage above and beyond helping us in the case of small number of instances?

  Assume that we have multiple fMRI datasets, with each dataset coming from a subject participating in a particular study. Let us say that we have an infinite number of instances in each dataset. Then intuitively, from a predictive standpoint, there is nothing to be gained in trying to jointly analyze these datasets together, because we have a complete characterization of the uncertainty in each dataset due to the infinite number of instances. This suggests that the answer to the above question is NO. Nonetheless, it is still desirable to have method that can jointly analyze multiple fMRI datasets. One reason is that in the real world, we are faced with the problem of only a limited number of instances available in fMRI datasets, especially relative to the number of features present in the data. A method to integrate multiple fMRI datasets would be useful because it can be considered as a way to increase the number of instances available to fit a predictive model, leading to a better model. In addition, such a method can potentially give us information about similarities and differences that exist in the various fMRI datasets. In turn, that can lead us toward a computational model of the brain that can account for both multiple subjects and multiple studies.

Based on what follows the second question, even though there might not be any benefit—in terms of predictive accuracy—in having a method that can integrate multiple fMRI datasets compared to a method that works on only a single fMRI dataset given a sufficient number of instances, it is still worthwhile to pursue a method capable of integrating multiple fMRI datasets. And in light of the first question, in this chapter we show that we can have a method to integrate multiple fMRI datasets that does not require that the multiple fMRI datasets belong to the same feature space. In particular, we work within the framework of the linear factor analysis.

3.1 Linear Factor Analysis Framework

In the linear factor analysis framework, it is assumed that there are factors underlying a dataset, and the data is generated as a linear combination of these underlying factors. More formally, if there are K underlying factors, a data instance $\mathbf{y}_i$ with dimension D_Y (in other words, $\mathbf{y}_i$ is a column vector of D_Y elements) is generated by the underlying factors as follows:

$$\mathbf{y}_i = \mathbf{W}\mathbf{z}_i + \boldsymbol{\epsilon}_i. \tag{3.1}$$

$\mathbf{z}_i$ is the column vector containing the K factors ($\mathbf{z}_i$ contains K elements), and $\mathbf{W}$ is the $D_Y \times K$ factor loading matrix, specifying how the data is generated from the factors. We also have a noise term $\boldsymbol{\epsilon}_i$, such that $\mathbb{E}[\boldsymbol{\epsilon}_i] = \mathbf{0}$. The elements of $\mathbf{z}_i$ can take any real values.

The discussion above applies to data from a single dataset. We can extend it to model multiple datasets by assuming that all the datasets share the same factors and have matching instances, the latter of which

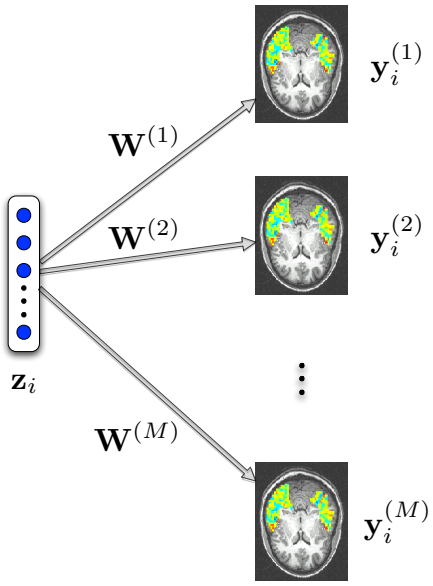


Figure 3.1.1: An application of the linear factor analysis framework to analyze multiple fMRI datasets jointly.

means that we can match instance i in one dataset to instance i in all the other datasets. More formally, assuming that we have M datasets, we can write this as

$$\mathbf{y}_i^{(m)} = \mathbf{W}^{(m)}\mathbf{z}_i + \boldsymbol{\epsilon}_i^{(m)}, 1 \leq m \leq M. \tag{3.2}$$

We can see that for instance i, $\mathbf{z}_i$ is shared across datasets, while each dataset has its own factor loading matrix $\mathbf{W}^{(m)}$. Also note that we can cast the multiple-dataset FA as a single-dataset FA by writing equation (3.2) in the form of equation (3.1), with

$$\mathbf{y}_i = \begin{bmatrix} \mathbf{y}_i^{(1)} \\ \mathbf{y}_i^{(2)} \\ \vdots \\ \mathbf{y}_i^{(M)} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \mathbf{W}^{(1)} \\ \mathbf{W}^{(2)} \\ \vdots \\ \mathbf{W}^{(M)} \end{bmatrix}, \quad \boldsymbol{\epsilon}_i = \begin{bmatrix} \boldsymbol{\epsilon}_i^{(1)} \\ \boldsymbol{\epsilon}_i^{(2)} \\ \vdots \\ \boldsymbol{\epsilon}_i^{(M)} \end{bmatrix}. \tag{3.3}$$

From the above formulation, note that all the datasets share the K factors, but there is no restriction that the dimension of each dataset, i.e. the length of each vector $\mathbf{y}_i^{(m)}$, be the same. We can let each data instance vector $\mathbf{y}_i^{(m)}$ be a column vector with $D_{Y^{(m)}}$ elements, potentially different for different m. This means that the factor loading matrix $\mathbf{W}^{(m)}$ needs to be of dimension $D_{Y^{(m)}} \times K$, but this is not a problem since each dataset has its own unique factor loading matrix, and all the datasets can still share the same factor instance $\mathbf{z}_i$. Hence, there is no need that all the datasets be in a common feature space, which addresses the first question above.

In addition, usually we want the number of factors K to be much lower than the dimension of each dataset $D_{Y^{(m)}}$. across the datasets. This means that we can consider the above formulation as a dimensionality reduction method, with the constraint that the reduced dimensions are common for all the datasets.

How does the linear factor analysis framework apply to the fMRI data? One setting, shown in figure 3.1.1, is where each variable $\mathbf{y}_i^{(m)}$ represents the fMRI activations for a particular instance i (e.g. trial) coming from a particular subject m; by applying the linear factor analysis framework to fMRI data coming from multiple subjects and/or studies, we can then find factors common across the fMRI datasets, which for a particular instance i will be contained in the variable $\mathbf{z}_i$. The projection represented by the loading matrix

$\mathbf{W}^{(m)}$ will capture how the factors project to the actual fMRI data for each subject m; $\mathbf{W}^{(m)}$ will give an indication about the variations of the fMRI data across subject and/or studies. This is a straightforward application of the linear factor analysis framework to jointly analyze fMRI data from multiple subjects and/or studies, and in chapter 4, we describe examples of this kind of application. Besides learning common factors across the fMRI data, the linear factor analysis framework can also be used to learn factors common across the fMRI datasets and other kinds of datasets. An example of this kind of application is described in chapter 5, where we consider including corpus and behavioral data in addition to fMRI data to learn common factors.

We have not yet discussed how we can estimate the parameters of the factor analysis model from the data, namely the factor loading matrices $\mathbf{W}^{(m)}$ and the factor score vectors $\mathbf{z}_i$. As formulated up to this point, the specification of the model is too general to lead to unique estimates for the parameters of the factor analysis model. This thesis focuses on a couple of approaches for learning these parameters. The first approach is to estimate these parameters using principal component analysis, described in section 3.2. This section follows the development of PCA in Johnson and Wichern (2002). The second approach considered in this thesis is the canonical correlation analysis (CCA), which is described in section 3.3. The materials in this section are based on presentations in Bach and Jordan (2002); Hardoon et al. (2004); Shawe-Taylor and Cristianini (2004). In section 3.4, we discuss briefly two other methods that can be used for estimating the parameters of multiple-dataset factor analysis model. All the above sections describe existing methods and are written in the style of a tutorial. All these methods assume that the different datasets to be analyzed jointly have matching instances. In section 3.6, we describe an original contribution, based on the idea of imputing missing data, to allow the above methods to be applied in the case where the datasets have some non-matching instances. The presentation of the results of applying these approaches to real fMRI data involving multiple subjects and studies is in chapters 4 and 5.

3.2 Principal component analysis (PCA) and/or singular value decomposition (SVD)

Let us consider a particular dataset represented as a matrix $\mathbf{X}(D \times N)$; here there is a row for each feature of the data and there is a column for each instance in the data. Let us also assume that each column/instance vector $\mathbf{x}_i$ of $\mathbf{X}$ satisfies $\mathbb{E}[\mathbf{x}_i] = \mathbf{0}$ and $\mathbb{E}[\mathbf{x}_i \mathbf{x}_i^T] = \mathbf{\Sigma}$ for some positive-definite $(D \times D)$ matrix $\mathbf{\Sigma}$. The principal component analysis (PCA) method (Pearson (1901); Hotelling (1933)) tries to find the linear combination of the dimensions such that the variance of the linear combination is maximized. More formally, we find the $D \times 1$ vector $\mathbf{w}$ that solves the following optimization problem:

$$\max_{\mathbf{w}} \mathrm{Var}(\mathbf{w}^T \mathbf{X}) = \mathbf{w}^T \mathbf{X} \mathbf{X}^T \mathbf{w} = \mathbf{w}^T \mathbf{C}_{xx} \mathbf{w}., \tag{3.4}$$

where $\mathbf{C}_{xx} = \mathbf{X}\mathbf{X}^T$. Because the solution of the above optimization is invariant to scaling and to avoid the resulting indeterminacy, a convention is to constrain the vector $\mathbf{w}$ to be a normal vector, i.e. it satisfies the property

$$\mathbf{w}^T \mathbf{w} = 1. \tag{3.5}$$

The solution of the above optimization problem yields the first principal component. Subsequent principal components can be found by imposing the same optimization problem (3.4) and the constraint (3.5), with the additional constraint that the covariance of the linear combination with any of the previous principal components is zero. In other words, for the i-th principal component, we add the constraint

$$\mathrm{Cov}(\mathbf{w}^T \mathbf{X}, \mathbf{w}_k^T \mathbf{X}) = 0, \tag{3.6}$$

where $\mathbf{w}_k$ corresponds to the k-th principal component, with $k < i$.

It turns out (see for instance Result 8.1 in Johnson and Wichern (2002)) that the vectors $\mathbf{w}$'s are eigenvectors of the eigenproblem

$$\mathbf{C}_{xx}\mathbf{w} = \lambda\mathbf{w}. \tag{3.7}$$

If we consider K principal components, we can group the vectors $\mathbf{w}$'s into a $D \times K$ matrix $\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_K]$. $\mathbf{W}$ has orthonormal columns, since each column is a normalized eigenvector. If we order the eigenvectors based on the sorted descending eigenvalue, then because of the above constraints, we have $\mathbf{w}_1^T\mathbf{X}\mathbf{X}^T\mathbf{w}_1 \geq \cdots \geq \mathbf{w}_K^T\mathbf{X}\mathbf{X}^T\mathbf{w}_K$ and for $k \neq l, \mathbf{w}k^T\mathbf{X}\mathbf{X}^T\mathbf{w}_l = 0$.

For a particular instance vector $\mathbf{x}_i$, let $\mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i$. $\mathbf{z}_i$ is a column vector with K elements, and since the columns of $\mathbf{W}$ are orthonormal, the following relationship holds:

$$\mathbf{x}_i = \mathbf{W}\mathbf{z}_i. \tag{3.8}$$

This equation shares the form of (3.1). Therefore, PCA gives a solution to the factor analysis model with one dataset, and in the case of multiple datasets, we can apply the concatenation of the different datasets following (3.3) and apply PCA to obtain a solution to the multiple-dataset factor analysis model.

An approach to find the solution of the factor analysis model (3.1) closely related to PCA is to use the singular value decomposition (SVD). For a general $M \times N$ matrix $\mathbf{A}$, its SVD is a decomposition $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$, where the matrix of left singular vectors $\mathbf{U}$ is an $M \times M$ orthogonal matrix, $\mathbf{S}$ is a $M \times N$ diagonal matrix, and the matrix of right singular vectors $\mathbf{V}$ is an $N \times N$ orthogonal matrix. When applying the SVD to the $D \times N$ dataset matrix $\mathbf{X}$ specified above, we obtain a decomposition $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$. The form of the SVD of $\mathbf{X}$ is similar to the form of the factor analysis model (3.1), where we take the factor loading matrix $\mathbf{W}$ to be a subset of the columns of either $\mathbf{U}$ or $\mathbf{U}\mathbf{S}$; the corresponding factor score matrix in this is the corresponding subset of the rows of either $\mathbf{S}\mathbf{V}^T$ or $\mathbf{V}^T$.

We mentioned that SVD is closely related to PCA. In fact, in the case where we select as the factor analysis loading matrix $\mathbf{W}$ the first K columns of the matrix of left singular vectors $\mathbf{U}$, we have the PCA model with K factors. This can be seen by noting that both the columns of the PCA loading matrix and the first K columns of $\mathbf{U}$ are the eigenvectors corresponding to the K largest eigenvalues of $\mathbf{X}\mathbf{X}^T = \mathbf{C}_{xx}$.

3.3 Canonical correlation analysis

Canonical correlation analysis (CCA) was first proposed in Hotelling (1936) as way to model the relationships between two related datasets. These relationships are characterized by what is called *canonical correlation components* or *canonical variates*. Let us formulate this more formally. Let the datasets be represented by matrices $\mathbf{X}$ and $\mathbf{Y}$, with dimensions $D_X \times N$ and $D_Y \times N$, respectively, where D_X (D_Y) denotes the number of dimensions in datasets $\mathbf{X}$ ($\mathbf{Y}$), and there are N instances in both datasets . Here we assume that both data matrices have mean zero across instances, or in other words, each column in each matrix has mean zero. The first canonical variate corresponds to projections $\mathbf{a_X} = \mathbf{w_X}^T\mathbf{X}$ and $\mathbf{a_Y} = \mathbf{w_Y}^T\mathbf{Y}$ such that $\mathbf{a_X}$ and $\mathbf{a_Y}$ are maximally correlated. The vectors $\mathbf{w_X}$ and $\mathbf{w_Y}$ are referred to as the *CCA loadings* for the first canonical variate. Note that the correlation of two scalar random variables U and V is defined as

$$\mathrm{Corr}(U, V) = \frac{\mathrm{Cov}(U, V)}{\sqrt{\mathrm{Var}(U)}\sqrt{\mathrm{Var}(V)}}. \tag{3.9}$$

In the CCA case, given the assumption that the data matrices have mean zero, the (sample) covariance of the projections of $\mathbf{X}$ and $\mathbf{Y}$ is given by $\mathbf{a_X}^T\mathbf{a_Y}$, and the variances of the projections are given by $\mathbf{a_X}^T\mathbf{a_X}$ and $\mathbf{a_Y}^T\mathbf{a_Y}$. Therefore, we can cast the problem of finding the first canonical variate as the following optimization problem:

$$\max_{\mathbf{w_X}, \mathbf{w_Y}} \frac{\mathbf{w_X}^T\mathbf{X}\mathbf{Y}^T\mathbf{w_Y}}{\sqrt{\mathbf{w_X}^T\mathbf{X}\mathbf{X}^T\mathbf{w_X}}\sqrt{\mathbf{w_Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y}}}. \tag{3.10}$$

Let $\mathbf{w}^*_\mathbf{X}$ and $\mathbf{w}^*_\mathbf{Y}$ be solutions to the above optimization problem. If we scale both $\mathbf{w}^*_\mathbf{X}$ and $\mathbf{w}^*_\mathbf{Y}$ by the same factor κ, we obtain

$$\begin{aligned}\frac{(\kappa\mathbf{w}_{\mathbf{X}}^*)^T\mathbf{X}\mathbf{Y}^T(\kappa\mathbf{w}_{\mathbf{Y}}^*)}{\sqrt{(\kappa\mathbf{w}_{\mathbf{X}}^*)^T\mathbf{X}\mathbf{X}^T(\kappa\mathbf{w}_{\mathbf{X}}^*)}\sqrt{(\kappa\mathbf{w}_{\mathbf{Y}}^*)^T\mathbf{Y}\mathbf{Y}^T(\kappa\mathbf{w}_{\mathbf{Y}}^*)}} &= \frac{\kappa^2\mathbf{w}_{\mathbf{X}}^{*T}\mathbf{X}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}}^*}{\kappa^2\sqrt{\mathbf{w}_{\mathbf{X}}^{*T}\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}}^*}\sqrt{\mathbf{w}_{\mathbf{Y}}^{*T}\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}}^*}} \\ &= \frac{\mathbf{w}_{\mathbf{X}}^{*T}\mathbf{X}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}}^*}{\sqrt{\mathbf{w}_{\mathbf{X}}^{*T}\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}}^*}\sqrt{\mathbf{w}_{\mathbf{Y}}^{*T}\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}}^*}},\end{aligned}$$

which means that the solutions to the optimization problem are invariant to scaling. For this reason, a standard convention is to find the solution such that the variances of $\mathbf{a}_{\mathbf{X}}$ and $\mathbf{a}_{\mathbf{Y}}$ are equal to one. This leads to the following modified optimization problem:

$$\begin{aligned}\max_{\mathbf{w}_{\mathbf{X}},\mathbf{w}_{\mathbf{Y}}} \quad & \mathbf{w}_{\mathbf{X}}{}^T\mathbf{X}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} \\ & \text{subject to} \\ & \mathbf{w}_{\mathbf{X}}{}^T\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}} = 1 \\ & \mathbf{w}_{\mathbf{Y}}{}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} = 1 \quad .\end{aligned} \tag{3.11}$$

There are typically more than one pair of canonical variates. The subsequent canonical variates are found using the same optimization problem with the additional constraints that they are orthogonal to the previous canonical variates. So if there are K canonical variates, we solve the following optimization problem:

$$\begin{aligned}\max_{\mathbf{w}_{\mathbf{X}}{}^{(1)},\mathbf{w}_{\mathbf{Y}}{}^{(1)},\cdots,\mathbf{w}_{\mathbf{X}}{}^{(K)},\mathbf{w}_{\mathbf{Y}}{}^{(K)}} \quad & (\mathbf{w}_{\mathbf{X}}{}^{(k)})^T\mathbf{X}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}}{}^{(k)}, \qquad 1 \le k \le K \\ & \text{subject to} \\ & (\mathbf{w}_{\mathbf{X}}{}^{(k)})^T\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}}{}^{(k)} = 1, \quad 1 \le k \le K \\ & (\mathbf{w}_{\mathbf{Y}}{}^{(k)})^T\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}}{}^{(k)} = 1, \quad 1 \le k \le K \\ & (\mathbf{w}_{\mathbf{X}}{}^{(k)})^T\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}}{}^{(l)} = 0, \quad 1 \le k \le K, 1 \le l \le K, k \ne l \\ & (\mathbf{w}_{\mathbf{Y}}{}^{(k)})^T\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}}{}^{(l)} = 0, \quad 1 \le k \le K, 1 \le l \le K, k \ne l.\end{aligned} \tag{3.12}$$

We now find the solution to the CCA optimization problem. We consider the optimization problem in equation (3.11). The Lagrangian for that optimization problem is given by

$$\mathcal{L}(\mathbf{w}_{\mathbf{X}}, \mathbf{w}_{\mathbf{Y}}, \lambda_1, \lambda_2) = \mathbf{w}_{\mathbf{X}}{}^T\mathbf{X}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} + \lambda_1(\mathbf{w}_{\mathbf{X}}{}^T\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}} - 1) + \lambda_2(\mathbf{w}_{\mathbf{Y}}{}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} - 1). \tag{3.13}$$

From the theory of Lagrange multipliers, we know that the solution of the optimization problem (3.11) has to satisfy the condition that at the solution, the gradient $\nabla\mathcal{L}(\mathbf{w}_{\mathbf{X}}, \mathbf{w}_{\mathbf{Y}}, \lambda_1, \lambda_2)$ has to be the zero vector. This means that all of $\frac{\partial\mathcal{L}}{\partial\mathbf{w}_{\mathbf{X}}}, \frac{\partial\mathcal{L}}{\partial\mathbf{w}_{\mathbf{Y}}}, \frac{\partial\mathcal{L}}{\partial\lambda_1}, \frac{\partial\mathcal{L}}{\partial\lambda_2}$ have to be zero. Let us consider the first two partial derivatives:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_{\mathbf{X}}} = \mathbf{X}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} + \lambda_1\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}} = \mathbf{0} \tag{3.14}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}_{\mathbf{Y}}} = \mathbf{Y}\mathbf{X}^T\mathbf{w}_{\mathbf{X}} + \lambda_2\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} = \mathbf{0} \tag{3.15}$$

Let us rewrite these two equations as

$$\mathbf{w}_{\mathbf{X}}{}^T\mathbf{X}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} + \lambda_1\mathbf{w}_{\mathbf{X}}{}^T\mathbf{X}\mathbf{X}^T\mathbf{w}_{\mathbf{X}} = 0 \tag{3.16}$$

$$\mathbf{w}_{\mathbf{Y}}{}^T\mathbf{Y}\mathbf{X}^T\mathbf{w}_{\mathbf{X}} + \lambda_2\mathbf{w}_{\mathbf{Y}}{}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w}_{\mathbf{Y}} = 0 \tag{3.17}$$

Since $\mathbf{w_X}^T\mathbf{X}\mathbf{Y}^T\mathbf{w_Y} = \mathbf{w_Y}^T\mathbf{Y}\mathbf{X}^T\mathbf{w_X}$, when we subtract the second equation from the first, we obtain

$$\lambda_1\mathbf{w_X}^T\mathbf{X}\mathbf{X}^T\mathbf{w_X} - \lambda_2\mathbf{w_Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y} = 0, \tag{3.18}$$

and since due to the constraints, $\mathbf{w_X}^T\mathbf{X}\mathbf{X}^T\mathbf{w_X} = \mathbf{w_Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y} = 1$, we have $\lambda_1 = \lambda_2 = -\lambda$ for some scalar real value λ. With this in mind, we revisit equations (3.14) and (3.15), substituting λ for λ_1 and λ_2:

$$\begin{aligned} \mathbf{X}\mathbf{Y}^T\mathbf{w_Y} - \lambda\mathbf{X}\mathbf{X}^T\mathbf{w_X} &= \mathbf{0} && (3.19)\\ \mathbf{Y}\mathbf{X}^T\mathbf{w_X} - \lambda\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y} &= \mathbf{0}, && (3.20) \end{aligned}$$

or

$$\begin{aligned} \mathbf{X}\mathbf{Y}^T\mathbf{w_Y} &= \lambda\mathbf{X}\mathbf{X}^T\mathbf{w_X} && (3.21)\\ \mathbf{Y}\mathbf{X}^T\mathbf{w_X} &= \lambda\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y}. && (3.22) \end{aligned}$$

We can rewrite these equations as

$$\begin{pmatrix} \mathbf{0} & \mathbf{X}\mathbf{Y}^T \\ \mathbf{Y}\mathbf{X}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w_X} \\ \mathbf{w_Y} \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{X}\mathbf{X}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}\mathbf{Y}^T \end{pmatrix} \begin{pmatrix} \mathbf{w_X} \\ \mathbf{w_Y} \end{pmatrix}. \tag{3.23}$$

This is a symmetric generalized eigenvalue problem $\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}$ with positive definite $\mathbf{B}$, for which there exist robust algorithms to find the solutions (see for instance, Demmel (1997)). A solution of (3.23) with non-zero eigenvalue also makes the gradient of the Lagrangian in (3.13) to be the zero vector, since

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\lambda_1} &= -(\mathbf{w_X}^T\mathbf{X}\mathbf{X}^T\mathbf{w_X} - 1) && (3.24)\\ \frac{\partial\mathcal{L}}{\partial\lambda_2} &= -(\mathbf{w_Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y} - 1), && (3.25) \end{aligned}$$

which are zero when the constraints are satisfied. So a necessary condition for a solution to the optimization problem (3.11) is that it has to satisfy the generalized eigenvalue problem (3.23). Furthermore, revisiting equation (3.21), we note that if we premultiply it by $\mathbf{w_X}^T$, we obtain

$$\mathbf{w_X}^T\mathbf{X}\mathbf{Y}^T\mathbf{w_Y} = \lambda\mathbf{w_X}^T\mathbf{X}\mathbf{X}^T\mathbf{w_X} = \lambda, \tag{3.26}$$

and since we are maximizing $\mathbf{w_X}^T\mathbf{X}\mathbf{Y}^T\mathbf{w_Y}$, the solution to (3.11) is the one associated with the largest eigenvalue λ corresponding to the generalized eigenvalue problem (3.23).

Note also that typically an algorithm for solving a symmetric generalized eigenvalue problem $\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}$ with $\mathbf{B}$ positive definite finds the solutions such that two unique eigenvectors $\mathbf{x_1}$ and $\mathbf{x_2}$ satisfy the conditions $\mathbf{x_1}^T\mathbf{B}\mathbf{x_2} = 0$ (Demmel (1997)). This means the solutions to (3.23) also satisfy the constraints of the more general optimization problem (3.12). This means that we can use (3.23) to obtain all canonical variates simultaneously, with the first K canonical variates corresponding to the solutions with the K largest eigenvalues.

3.3.1 CCA as a factor analysis model

Before delving into other aspects of CCA, let us look at its connection with the factor analysis model described in the beginning of the chapter. Let us consider the CCA with K canonical variates. The standard CCA can be viewed as a version of the factor analysis model for two datasets, by viewing the canonical variates as factors in the factor analysis model.

More specifically, for $M = 2$ and assuming that there are N instances, we combine the data instances in equation (3.2) into data matrices $\mathbf{Y}^{(1)}$ ($D_{Y^{(1)}} \times N$) and $\mathbf{Y}^{(2)}$ ($D_{Y^{(2)}} \times N$). We make these matrices correspond

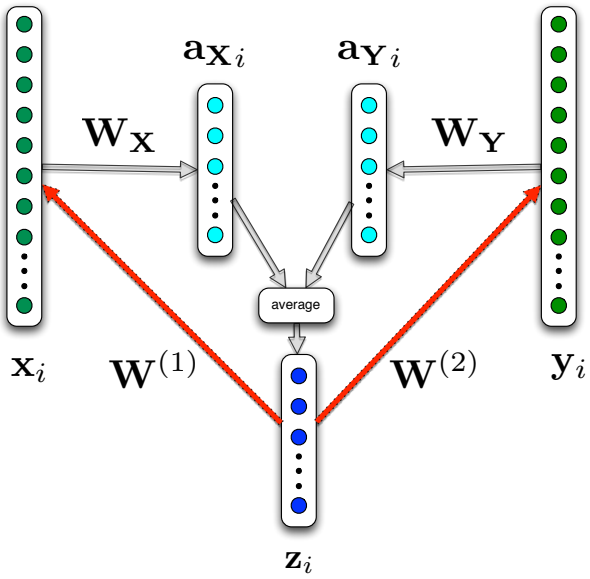


Figure 3.3.1: Illustration of CCA as a factor analysis model. The red arrows labeled with $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$ show the direction of the original factor analysis model as formulated in (3.2), which is slightly different from how CCA is formulated, shown in the rest of the figure.

to the $\mathbf{X}$ and $\mathbf{Y}$ matrices, respectively, in the CCA model. Next we combine the CCA loading vectors $\mathbf{w}_\mathbf{X}$ and $\mathbf{w}_\mathbf{Y}$ for the K canonical variates into CCA loading matrices $\mathbf{W}_\mathbf{X}$ ($K \times D_X$) and $\mathbf{W}_\mathbf{Y}$ ($K \times D_Y$). In the factor analysis model, we have the relationship $\mathbf{Y}^{(1)} = \mathbf{W}^{(1)}\mathbf{Z}$ ($\mathbf{Y}^{(2)} = \mathbf{W}^{(2)}\mathbf{Z}$), while in the CCA model, we have the relationship $\mathbf{W}_\mathbf{X}\mathbf{X} = \mathbf{A}_\mathbf{X}$ ($\mathbf{W}_\mathbf{Y}\mathbf{Y} = \mathbf{A}_\mathbf{Y}$). So the matrix $\mathbf{W}_\mathbf{X}$ ($\mathbf{W}_\mathbf{Y}$) is roughly the inverse of the matrix $\mathbf{W}^{(1)}$ ($\mathbf{W}^{(2)}$). However, since these matrices are not square, we need to use a generalized notion of inverse, such as the Moore-Penrose pseudoinverse. An illustration of what is just described is shown in figure 3.3.1, for a particular instance i. In this figure, because we consider a particular instance, the column vectors $\mathbf{z}_i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{a}_{\mathbf{X}i}, \mathbf{a}_{\mathbf{Y}i}$ replace the corresponding matrices $\mathbf{Z}, \mathbf{X}, \mathbf{Y}, \mathbf{A}_\mathbf{X}$, and $\mathbf{A}_\mathbf{Y}$.

The figure brings into light one notable difference between the factor analysis and the CCA models. While the factor scores are shared across datasets in the factor analysis model, in the CCA model, we have different albeit maximally correlated factors (represented by the canonical variates) for the two datasets. We see in figure 3.3.1 a way to relate the factor score matrix $\mathbf{Z}$ with the canonical variates matrices $\mathbf{A}_\mathbf{X}$ and $\mathbf{A}_\mathbf{Y}$. As shown in the figure, for our experiments, we use a heuristic to reduce the dataset-specific factors in CCA into common factors by taking the sample mean across datasets. Note also that our CCA model is also limited compared to the factor analysis model (3.2) in the sense that it can accept only two datasets, compared to the arbitrary number of datasets modeled in (3.2). Later, we consider an extension of CCA that can accept multiple datasets.

3.3.2 Kernel and regularized versions

Let us revisit the objective function (3.10) used in the optimization leading to the first canonical variates:

$$\max_{\mathbf{w}_\mathbf{X}, \mathbf{w}_\mathbf{Y}} \frac{\mathbf{w}_\mathbf{X}{}^T \mathbf{X}\mathbf{Y}^T \mathbf{w}_\mathbf{Y}}{\sqrt{\mathbf{w}_\mathbf{X}{}^T \mathbf{X}\mathbf{X}^T \mathbf{w}_\mathbf{X}}\sqrt{\mathbf{w}_\mathbf{Y}{}^T \mathbf{Y}\mathbf{Y}^T \mathbf{w}_\mathbf{Y}}}.$$

Let $\mathbf{w}_\mathbf{X} = \mathbf{X}\boldsymbol{\alpha}_X$ and $\mathbf{w}_\mathbf{Y} = \mathbf{Y}\boldsymbol{\alpha}_Y$, and let $\mathbf{K}_\mathbf{x} = \mathbf{X}^T\mathbf{X}$ and $\mathbf{K}_\mathbf{y} = \mathbf{Y}^T\mathbf{Y}$. We can recast the optimization problem as one over $\boldsymbol{\alpha}_X$ and $\boldsymbol{\alpha}_Y$ by modifying the objective function:

$$\max_{\boldsymbol{\alpha}_X, \boldsymbol{\alpha}_Y} \frac{\boldsymbol{\alpha}_X^T \mathbf{K_x} \mathbf{K_y} \boldsymbol{\alpha}_Y}{\sqrt{\boldsymbol{\alpha}_X^T \mathbf{K_x}^2 \boldsymbol{\alpha}_X} \sqrt{\boldsymbol{\alpha}_Y^T \mathbf{K_y}^2 \boldsymbol{\alpha}_Y}}. \tag{3.27}$$

This is the dual formulation of CCA, as opposed to the primal formulation (3.10). The matrices $\mathbf{K_x}$ and $\mathbf{K_y}$ are $N \times N$ Gram matrices whose entries are dot products of each instance vector from $\mathbf{X}$ and $\mathbf{Y}$, respectively. More specifically, given $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_K]$ where each $\mathbf{x}_k$ is a column vector, an entry at row i and column j of $\mathbf{K_x}$ is given by $\mathbf{K_x}(i,j) = \mathbf{x}_i^T \mathbf{x}_j$, and similarly for $\mathbf{Y}$. Furthermore, $\mathbf{K_x}$ and $\mathbf{K_y}$ can also be matrices of dot products in higher-dimensional features spaces: $\mathbf{K_x}(i,j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$; often this dot product can be represented by a kernel function κ of the pair of data instance vectors: $\mathbf{K_x}(i,j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j)$. This allows us to perform CCA nonlinearly, and since the dimensions of $\mathbf{K_x}$ and $\mathbf{K_y}$ do not grow with growing dimensions, the complexity of the computation is similar to the computational complexity of the linear CCA. In this thesis, however, since we are dealing with high-dimensional data—for which we have many more features than instances—it is relatively easy to over-fit the data with complex models such as nonlinear versions of CCA, so we focus mainly on the basic linear CCA.

Next we consider how to solve for (3.27). To obtain the solutions of (3.27), we note that its form is very similar to (3.10). In fact, following the steps outlined for the primal formulation, we can show that the above optimization problem is equivalent to the following (symmetric) generalized eigenvalue problem:

$$\begin{pmatrix} \mathbf{0} & \mathbf{K_x K_y} \\ \mathbf{K_y K_x} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_X \\ \boldsymbol{\alpha}_Y \end{pmatrix} = \rho \begin{pmatrix} \mathbf{K_x}^2 & \mathbf{0} \\ \mathbf{0} & \mathbf{K_y}^2 \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_X \\ \boldsymbol{\alpha}_Y \end{pmatrix}. \tag{3.28}$$

The form of the generalized eigenvalue problem (3.28) is very similar to the form in (3.23). A notable difference is that, the matrices involved in (3.23) have dimensions $(D_X + D_Y) \times (D_X + D_Y)$, while those involved in (3.28) have dimensions $2N \times 2N$. If $D_X \gg N$ (or $D_Y \gg N$), as is mostly the case with fMRI data, then (3.28) involves smaller matrices compared to (3.23), so solving the dual is more efficient.

On the other hand, the fact that $D_X \gg N$ (or $D_Y \gg N$) potentially gives rise to over-fitting. This can be seen clearly in the dual formulation. The generalized eigenvalue problem (3.28) is equivalent to the system of equations

$$\begin{aligned} \mathbf{K_x K_y} \boldsymbol{\alpha}_Y &= \rho \mathbf{K_x}^2 \boldsymbol{\alpha}_X & (3.29) \\ \mathbf{K_y K_x} \boldsymbol{\alpha}_X &= \rho \mathbf{K_y}^2 \boldsymbol{\alpha}_Y. & (3.30) \end{aligned}$$

Let us consider (3.29). If $\mathbf{K_x}$ is invertible, we have

$$\begin{aligned} \mathbf{K_x K_y} \boldsymbol{\alpha}_Y &= \rho \mathbf{K_x}^2 \boldsymbol{\alpha}_X & (3.31) \\ \mathbf{K_x}^{-1} \mathbf{K_x K_y} \boldsymbol{\alpha}_Y &= \rho \mathbf{K_x}^{-1} \mathbf{K_x}^2 \boldsymbol{\alpha}_X & (3.32) \\ \mathbf{K_y} \boldsymbol{\alpha}_Y &= \rho \mathbf{K_x} \boldsymbol{\alpha}_X. & (3.33) \end{aligned}$$

Substituting into (3.30), we obtain

$$\begin{aligned} \mathbf{K_y K_x} \boldsymbol{\alpha}_X &= \rho \mathbf{K_y} \underbrace{\mathbf{K_y} \boldsymbol{\alpha}_Y}_{\rho \mathbf{K_x} \boldsymbol{\alpha}_X} & (3.34) \\ \mathbf{K_y K_x} \boldsymbol{\alpha}_X &= \rho^2 \mathbf{K_y K_x} \boldsymbol{\alpha}_X. & (3.35) \end{aligned}$$

This relationship is trivially true with $\rho = 1$. This means that the eigenvalues are all one, which implies that the correlations of all the pairs of the canonical variates are all one. It can be shown in a similar fashion that this is also the case when $\mathbf{K_y}$ is invertible. The fact that all the correlations are one is an indication that the model over-fits, and as mentioned above, the condition for over-fitting likely applies to fMRI data. To avoid over-fitting, we need to regularize the CCA model.

Vinod (1976) presents a regularization for the standard CCA similar to what is done in the ridge regression. Taking the primal formulation (3.10), Vinod (1976) extends it by modifying the terms in the denominator, adding a (scaled) dot product of the CCA loading vectors:

$$\max_{\mathbf{w_X},\mathbf{w_Y}} \frac{\mathbf{w_X}^T\mathbf{X}\mathbf{Y}^T\mathbf{w_Y}}{\sqrt{\mathbf{w_X}^T\mathbf{X}\mathbf{X}^T\mathbf{w_X}+\kappa_x\mathbf{w_X}^T\mathbf{w_X}}\sqrt{\mathbf{w_Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y}+\kappa_y\mathbf{w_Y}^T\mathbf{w_Y}}}. \tag{3.36}$$

Again, we can normalize each square-root term to equal to one, yielding the constrained optimization problem:

$$\max_{\mathbf{w_X},\mathbf{w_Y}} \quad \mathbf{w_X}^T\mathbf{X}\mathbf{Y}^T\mathbf{w_Y} \tag{3.37}$$

$$\text{subject to} \tag{3.38}$$

$$\mathbf{w_X}^T\mathbf{X}\mathbf{X}^T\mathbf{w_X}+\kappa_x\mathbf{w_X}^T\mathbf{w_X}=1 \tag{3.39}$$

$$\mathbf{w_Y}^T\mathbf{Y}\mathbf{Y}^T\mathbf{w_Y}+\kappa_y\mathbf{w_Y}^T\mathbf{w_Y}=1 \quad . \tag{3.40}$$

This is the canonical ridge model of Vinod (1976) in primal form. We can also obtain the dual form following the steps outlined previously:

$$\max_{\boldsymbol{\alpha}_X,\boldsymbol{\alpha}_Y} \frac{\boldsymbol{\alpha}_X^T\mathbf{K_x}\mathbf{K_y}\boldsymbol{\alpha}_Y}{\sqrt{\boldsymbol{\alpha}_X^T\mathbf{K_x}^2\boldsymbol{\alpha}_X+\kappa_x\boldsymbol{\alpha}_X\mathbf{K_x}\boldsymbol{\alpha}_X}\sqrt{\boldsymbol{\alpha}_Y^T\mathbf{K_y}^2\boldsymbol{\alpha}_Y+\kappa_y\boldsymbol{\alpha}_Y\mathbf{K_y}\boldsymbol{\alpha}_Y}}. \tag{3.41}$$

In this formulation of the canonical ridge model, the regularization coefficients κ_x and $\kappa_Y y$ range between 0 and ∞. For our experiments, we use an alternative formulation of the canonical ridge model with regularization coefficients τ_x and τ_y that range from 0 to 1 (Shawe-Taylor and Cristianini (2004)); in the dual form, this is formulated as

$$\max_{\boldsymbol{\alpha}_X,\boldsymbol{\alpha}_Y} \frac{\boldsymbol{\alpha}_X^T\mathbf{K_x}\mathbf{K_y}\boldsymbol{\alpha}_Y}{\sqrt{(1-\tau_x)\boldsymbol{\alpha}_X^T\mathbf{K_x}^2\boldsymbol{\alpha}_X+\tau_x\boldsymbol{\alpha}_X\mathbf{K_x}\boldsymbol{\alpha}_X}\sqrt{(1-\tau_y)\boldsymbol{\alpha}_Y^T\mathbf{K_y}^2\boldsymbol{\alpha}_Y+\tau_y\boldsymbol{\alpha}_Y\mathbf{K_y}\boldsymbol{\alpha}_Y}}. \tag{3.42}$$

We can also derive a generalized eigenvalue problem associated with the canonical ridge model. The one associated with (3.42) is given by

$$\begin{pmatrix} \mathbf{0} & \mathbf{K_x}\mathbf{K_y} \\ \mathbf{K_y}\mathbf{K_x} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \boldsymbol{\alpha}_X \\ \boldsymbol{\alpha}_Y \end{pmatrix} = \rho\begin{pmatrix} (1-\tau_x)\mathbf{K_x}^2+\tau_x\mathbf{K_x} & \mathbf{0} \\ \mathbf{0} & (1-\tau_y)\mathbf{K_y}^2+\tau_y\mathbf{K_y} \end{pmatrix}\begin{pmatrix} \boldsymbol{\alpha}_X \\ \boldsymbol{\alpha}_Y \end{pmatrix}. \tag{3.43}$$

So we can obtain the solutions for the canonical ridge model with roughly equivalent efficiency as for the solutions of the original CCA model.

3.3.3 Multiple datasets

In the discussion so far, we have considered the standard formulation of CCA which can be applied to only two datasets. However, the application that is the focus of the thesis on the analysis of fMRI data from multiple subjects and multiple studies, we usually have more than two datasets to analyze. We can make the problem amenable to the standard CCA by combining the different datasets into a single matrix, in the style of (3.3). Then CCA can be used if we want to analyze the fMRI data together with another kind of data, as we show in the next chapter. However, we also ask the question whether it is possible to extend CCA to accept more than two data matrices separately. We now consider this question.

One possible way to generalize CCA to more than two datasets is to start with the generalized eigenvalue problem form of the solution, given in equation (3.23) for the primal version of the standard CCA and in equation (3.43) for the regularized kernel CCA. We can extend these generalized eigenvalue problems in a straightforward manner to handle multiple datasets. In the primal form, this is given by

$$\begin{pmatrix} \mathbf{0} & \mathbf{C}_{12} & \cdots & \mathbf{C}_{1M} \\ \mathbf{C}_{21} & \mathbf{0} & \cdots & \mathbf{C}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{M1} & \cdots & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \vdots \\ \mathbf{w}_M \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{C}_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \vdots \\ \vdots \\ \mathbf{w}_M \end{pmatrix}, \tag{3.44}$$

i.e. on the left-hand side, we expand the matrix with zeros on the diagonal and cross-covariance terms on the off-diagonal, while on the right-hand side, we expand the matrix with the variance terms on the diagonal and zeros on the off-diagonal. This is equivalent to the optimization problem

$$\max_{\mathbf{w}_1,\cdots,\mathbf{w}_M} \quad \sum_{m'\neq m, 1\leq m,m'\leq M} \mathbf{w}_m^T \mathbf{X}_m \mathbf{X}_{m'}^T \mathbf{w}_{m'} \tag{3.45}$$

$$\text{subject to} \tag{3.46}$$

$$(1-\tau_m)\mathbf{w}_m^T \mathbf{X}_m \mathbf{X}_m^T \mathbf{w}_m + \tau_m \mathbf{w}_m^T \mathbf{w}_m \quad 1 \leq m \leq M \quad . \tag{3.47}$$

We can apply the same procedure to all the other forms of CCA we have considered so far. In particular, for the regularized dual version of CCA,

$$\begin{pmatrix} \mathbf{0} & \mathbf{K_1 K_2} & \cdots & \mathbf{K_1 K_M} \\ \mathbf{K_2 K_1} & \mathbf{0} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K_M K_1} & \cdots & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \vdots \\ \boldsymbol{\alpha}_M \end{pmatrix} \tag{3.48}$$

$$= \rho \begin{pmatrix} (1-\tau_1)\mathbf{K_1}^2 + \tau_1 \mathbf{K_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & (1-\tau_2)\mathbf{K_2}^2 + \tau_2 \mathbf{K_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & (1-\tau_M)\mathbf{K_M}^2 + \tau_M \mathbf{K_M} \end{pmatrix} \begin{pmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \vdots \\ \boldsymbol{\alpha}_M \end{pmatrix}, \tag{3.49}$$

which is equivalent to the optimization problem

$$\max_{\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_M} \quad \sum_{m'\neq m, 1\leq m,m'\leq M} \boldsymbol{\alpha}_m^T \mathbf{K}_m \mathbf{K}_{m'} \boldsymbol{\alpha}_{m'} \tag{3.50}$$

$$\text{subject to} \tag{3.51}$$

$$(1-\tau_m)\boldsymbol{\alpha}_m^T \mathbf{K}_m^2 \boldsymbol{\alpha}_m + \tau_m \boldsymbol{\alpha}_m^T \mathbf{K}_m \boldsymbol{\alpha}_m \quad 1 \leq m \leq M \quad . \tag{3.52}$$

As shown in (3.3), one way to analyze multiple datasets jointly within a factor analysis model is by concatenating the data matrices. In particular, if we have more than two datasets, we can still use the classical 2-way CCA by reducing the multiple data matrices to two matrices using (3.3). We now see how this compare with the multi-way CCA presented in this section.

We work in the context of the unregularized primal version. We assume that there are M datasets $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_M$, and we have the last $M-1$ datasets concatenated:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_M \end{bmatrix}. \tag{3.53}$$

We see that

$$\mathbf{X}_1\mathbf{Y}^T = [\mathbf{C}_{12}\cdots\mathbf{C}_{1M}] \tag{3.54}$$

$$\mathbf{Y}\mathbf{Y}^T = \begin{bmatrix} \mathbf{C}_{22} & \cdots & \mathbf{C}_{2M} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{M2} & \cdots & \mathbf{C}_{MM} \end{bmatrix}. \tag{3.55}$$

Plugging these into (3.23), we obtain

$$\begin{pmatrix} \mathbf{0} & \mathbf{C}_{12} & \cdots & \mathbf{C}_{1M} \\ \mathbf{C}_{21} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{M1} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{C}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_{22} & \cdots & \mathbf{C}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{C}_{M2} & \cdots & \mathbf{C}_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix}. \tag{3.56}$$

Unlike in (3.44), only the cross-covariance terms involving $\mathbf{X}_1$ are in the left-hand matrix, while the rest of the cross-covariance terms are in the right-hand matrix, causing the right-hand matrix to not be block-diagonal any more. In the next chapter, we see how this difference affects the predictive performance of the model.

The generalization of CCA considered in this section arises from a straightforward generalization of the associated generalized eigenvalue problem, but it is by no means the only one possible. We refer to Kettenring (1971) for other generalizations of CCA to multiple datasets, although for our experiments, we use only the version presented here.

3.3.4 Comparison of PCA and CCA

We now compare the PCA model when applied to multiple datasets and the CCA model. From our discussion, it is clear that the PCA solution produces one set of factors for all datasets while in the case of CCA we have one set of factors for each dataset. Furthermore, in the PCA case, the variance of the projection is maximized, while in the CCA the variance of the projection is constrained to be one (modulo some regularization). On the other hand, PCA has the constraint that the the loading vector $\mathbf{w}_k$ is orthonormal, while there is no such constraint in CCA.

We can also compare the eigenvalue problem associated with PCA involving multiple datasets with those of the CCA model. Letting

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_M \end{bmatrix}, \tag{3.57}$$

PCA's eigenvalue problem given in (3.7) then becomes

$$\begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \cdots & \mathbf{C}_{1M} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \cdots & \mathbf{C}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{M1} & \mathbf{C}_{M2} & \cdots & \mathbf{C}_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix}, \tag{3.58}$$

or equivalently

$$\begin{pmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} & \cdots & \mathbf{C}_{1M} \\ \mathbf{C}_{21} & \mathbf{C}_{22} & \cdots & \mathbf{C}_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{C}_{M1} & \mathbf{C}_{M2} & \cdots & \mathbf{C}_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{I}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{22} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_{MM} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_M \end{pmatrix}, \tag{3.59}$$

where each $\mathbf{I}_{mm}$ is an identity matrix with as many rows and columns as the number of rows of $\mathbf{X}_m$. Contrasting this with (3.44), we see that another way to see the difference between PCA and the multiple-dataset CCA, is to look into how the within-dataset covariance and cross-dataset covariance terms are grouped in a particular symmetric generalized eigenvalue problem: in the case of PCA, all these terms are put in the left matrix while the right matrix becomes the identity matrix, while in the case of the multiple-dataset CCA, the left matrix contains purely cross-dataset covariance terms, with the within-dataset covariance terms belonging to the right matrix.

3.4 Other factor analytic approaches

The factor analysis models specified in (3.2) are sufficiently general such that the solutions might not be unique. As we see above, PCA and the various CCA models give particular solutions to (3.2) by imposing particular assumptions on the form that the solutions can take. Now we take a look at a couple other ways solutions to (3.2) can be obtained. However, we do not perform experiments using the methods described in this section.

3.4.1 Generalized Singular Value Decomposition

It turns out that we can extend the SVD to more than two matrices. Given two matrices $\mathbf{A}(D_A \times N)$ and $\mathbf{B}(D_B \times N)$ with equal number of columns, there exists a unique generalized singular value decomposition (GSVD, Van Loan (1976); Paige and Saunders (1981)) of $\mathbf{A}$ and $\mathbf{B}$ given by

$$\begin{aligned} \mathbf{A} &= \mathbf{U}\mathbf{C}\mathbf{F}^T \\ \mathbf{B} &= \mathbf{V}\mathbf{S}\mathbf{F}^T, \end{aligned} \tag{3.60}$$

where $\mathbf{U}$ and $\mathbf{V}$ are $D_A \times D_A$ and $D_B \times D_B$ orthogonal matrices, $\mathbf{C}$ and $\mathbf{S}$ are non-negative diagonal matrices, and $\mathbf{F}$ is an $N \times N$ matrix. Similar to the SVD case, we can relate this to the factor analysis model, in particular for the 2-dataset case of (3.2). We might be tempted to follow the PCA case and use $\mathbf{U}$ and $\mathbf{V}$ as loading matrices; however, with this approach, we end up with different factor score matrices $\mathbf{C}\mathbf{F}^T$ and $\mathbf{S}\mathbf{F}^T$ for the two datasets, since $\mathbf{C}$ and $\mathbf{S}$ are different. The natural approach then is to use $\mathbf{U}\mathbf{C}$ and $\mathbf{V}\mathbf{S}$ as loading matrices for the two datasets, with $\mathbf{F}^T$ serving as the shared factor score matrix.

The diagonal matrices $\mathbf{C}$ and $\mathbf{S}$ contain the generalized singular values for each dataset on each of their diagonals. Each generalized singular value indicates the significance of the corresponding component or factor. Typically, the generalized singular values are ordered from large to small for one of the datasets, and from small to large for the other dataset. This implies that some factors are more significant for one dataset compared to the other and some other factors are roughly equally significant for both datasets. A more precise way to determine the relative significance for a particular factor in both datasets is by looking at the ratio of the corresponding generalized singular values, for instance, for factor k, the ratio $\mathbf{C}(k,k)/\mathbf{S}(k,k)$. In practice, the transformation

$$\theta_k = \arctan(\mathbf{C}(k,k)/\mathbf{S}(k,k)) - \pi/4, \tag{3.61}$$

is commonly used as a metric, the reason being that it converts the ratio to a roughly uniform range centered at zero, with the zero value, corresponding to $\mathbf{C}(k,k)/\mathbf{S}(k,k) = 1$, indicating that the corresponding factor is equally significant for both datasets.

As described in the previous paragraph, for GSVD a factor significant for one dataset might not be significant for the other dataset. This is in contrast with CCA, where each factor shares equal significance in both datasets. For this reason, unlike in the case of CCA where we can select the first few factors as the most significant, regardless of the datasets, in GSVD, the choice of a subset of factors depend on a particular dataset. This affects when we want to extrapolate from one dataset to the other dataset, meaning that when we have an instance of one dataset—let us say dataset $\mathbf{A}$—and we want to predict the corresponding

instance in the other dataset, $\mathbf{B}$ in this example. In this case, the extrapolation procedure should utilize factors significant for $\mathbf{B}$, but it is also important that the procedure includes factors that are significant for both $\mathbf{A}$ and $\mathbf{B}$, i.e. factors k with $\theta_k \approx 0$, so that the pertinent shared information is included in the formulation of the corresponding predicted instance for dataset $\mathbf{B}$. This brings the issue of choosing the threshold for θ_k, and to do this one set of guidelines are presented in Alter et al. (2003).

As with the classical CCA, GSVD works on only two matrices. It turns out that there is an extension of GSVD to more than two matrices, called the higher-order GSVD (HOGSVD, Ponnapalli et al. (2009)). Given data matrices $\mathbf{X}_1(D_1 \times N), \cdots, \mathbf{X}_M(D_M \times N)$ the HOGSVD decomposes them as follows:

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{U}_1\mathbf{C}_1\mathbf{F}^T && (3.62)\\ &\vdots && (3.63)\\ \mathbf{X}_M &= \mathbf{U}_M\mathbf{C}_M\mathbf{F}^T. && (3.64) \end{aligned}$$

The matrices involved are straightforward generalizations of the matrices involved in (3.60). Tying back to (3.2), we have $\mathbf{W}^{(m)} = \mathbf{U}_m\mathbf{C}_m, 1 \leq m \leq M$ and $\mathbf{Z} = \mathbf{F}^T$. In the HOGSVD case, the ratio of the generalized singular values can be used to determine the significance between a particular pair of datasets.

3.4.2 Orthogonal Factor Analysis

Another way to obtain solutions to (3.1) can be obtained by adding a few more assumptions to (3.1):

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{\epsilon}_i) &= \boldsymbol{\Psi}, \quad \boldsymbol{\Psi} \text{ diagonal} && (3.65)\\ \mathbb{E}[\mathbf{z}_i] &= \mathbf{0} && (3.66)\\ \mathrm{Cov}(\mathbf{z}_i) &= \mathbf{I}, \quad \text{the identity matrix.} && (3.67) \end{aligned}$$

In the literature, (3.1) with the above assumptions is commonly referred to as *factor analysis*, but since in this thesis we already use the term *factor analysis* for the general models (3.1) and (3.2), we refer to the above model ((3.1 with the additional assumptions) as the *orthogonal factor analysis* model, a term used in Johnson and Wichern (2002), since one of the assumptions is that the different factors are assumed to be orthogonal (in expectation). As with the PCA case, this can be applied to one dataset as in (3.1), or to multiple datasets by concatenating the data matrices as outlined in (3.3).

There are a couple of ways to obtain estimates of the parameters of this model. The first way, less commonly used in practice, is called the principal component method (Johnson and Wichern (2002)). In this method, we first compute the eigenvectors and eigenvalues of the sample covariance matrix of the data matrix $\mathbf{X}$. Let $(\hat{\lambda}_1, \hat{\mathbf{e}}_1), \cdots, (\hat{\lambda}_{D_X}, \hat{\mathbf{e}}_{D_X})$ be the eigenvalue-eigenvector pairs, ordered such that $\hat{\lambda}_1 \geq \cdots \geq \hat{\lambda}_{D_X}$. Then if K factors are assumed, for the factor loading matrix we have

$$\mathbf{W} = \left[\sqrt{\hat{\lambda}_1}\hat{\mathbf{e}}_1 \cdots \sqrt{\hat{\lambda_K}}\hat{\mathbf{e}}_K\right]. \tag{3.68}$$

Note that this is similar to what is done in the PCA case, described in section 3.2 above. The difference is that in the PCA case, in the loading matrix the eigenvectors are not scaled at all.

The second way to obtain the parameter estimate of the orthogonal factor analysis model is by adding yet more assumptions that $\mathbf{z}_i$ and $\boldsymbol{\epsilon}_i$ are normally distributed, and obtain the parameters that maximize the corresponding likelihood. This is the maximum likelihood estimation method. Because closed forms for the parameter estimates are not available, in practice, iterative techniques such as the EM algorithm (Dempster et al. (1977)) are used.

When we assume that $\mathbf{z}_i$ and $\boldsymbol{\epsilon}_i$ are normally distributed, equation (3.1) becomes an instance of what is referred to as the linear Gaussian models (Roweis and Ghahramani (1999)). Hence, the orthogonal factor analysis is a special case of the linear Gaussian models. It turns out that PCA also has some connection

with these linear Gaussian models. In particular, we can obtain PCA as the limit of another case of the linear Gaussian models (details can be found in Tipping and Bishop (1999); Roweis (1998); Roweis and Ghahramani (1999)). This makes explicit the connection between the orthogonal factor analysis and PCA.

One property of the orthogonal factor analysis is that the rotations of a particular set of solutions also produce valid solutions to the model. To see this, we note first that the covariance matrix $\mathbf{\Sigma}$ of the data can be written as $\Sigma = \mathbf{W}\mathbf{W}^T + \mathbf{\Psi}$. On the other hand, a rotation can be represented as a multiplication by an orthogonal matrix $\mathbf{R}$, and because of $\mathbf{R}$'s orthogonality, $\mathbf{R}\mathbf{R}^T = \mathbf{R}^T\mathbf{R} = \mathbf{I}$ the identity matrix. When we rotate the loading matrix $\mathbf{W}$, yielding $\mathbf{W}^* = \mathbf{W}\mathbf{R}$, we see that

$$\mathbf{W}^*(\mathbf{W}^*)^T + \mathbf{\Psi} = \mathbf{W}\mathbf{R}\mathbf{R}^T\mathbf{W}^T + \mathbf{\Psi} = \mathbf{W}\mathbf{W}^T + \mathbf{\Psi} = \mathbf{\Sigma}, \tag{3.69}$$

so the covariance matrix of the data can be accounted by the rotated factor loading matrix $\mathbf{W}^*$ as well as by the original factor loading matrix $\mathbf{W}$.

The fact that the parameters of the orthogonal factor analysis model are determined only up to some given rotation gives rise to the question of selecting an appropriate rotation after the estimation of the parameters. A popular choice is to use the varimax procedure (Kaiser (1957)), which selects the rotation that maximizes the variance of the loadings. More precisely, let $\mathbf{W}^* = \mathbf{W}\mathbf{R}$ be a particular rotated loading matrix rotated using the rotation matrix $\mathbf{R}$, and let

$$\hat{\mathbf{W}}^*(d,k) = \frac{\mathbf{W}^*(d,k)}{\sqrt{\sum_{k'=1}^{K} \mathbf{W}^*(d,k')}}, 1 \le d \le D_X, 1 \le k \le K. \tag{3.70}$$

The varimax procedure selects $\mathbf{R}$ that maximizes

$$\frac{1}{D_X}\sum_{k=1}^{K}\left(\sum_{d=1}^{D_X}\hat{\mathbf{W}}^*(d,k)^4 - \frac{1}{p}\left(\sum_{d=1}^{D_X}\hat{\mathbf{W}}^*(d,k)^2\right)^2\right). \tag{3.71}$$

In essence, the quantity to be maximized is the sum of variances of squared scaled loadings for all the factors (Johnson and Wichern (2002)). One effect of the procedure is to have a few large-magnitude loadings with the remaining loadings close to zero in each column. This helps the task of interpreting the factors, which is one reason why it is popular, although it remains to be seen whether it improves predictability as well.

3.5 Interlude

	PCA	CCA	GSVD	orthogonal FA
how multiple datasets are handled	concatenation	explicit	explicit	concatenation
order of factor importance	yes for all datasets	yes for all datasets	yes for individual datasets	no
shared factor scores	yes	no	yes	yes
unique solution	yes	yes	yes	no

Table 3.5.1: Comparison of the factor analysis estimation methods

We have seen four methods to obtain solutions within the context of the linear factor analysis framework given by (3.1) and (3.2). We group the major characteristics of these methods into four categories:

- how a method handles multiple datasets, whether through concatenation or explicit modeling
- whether a method produces an order of importance for the factors

- whether the factor scores produced by a method are shared across datasets
- whether the solution obtained by a method is unique

We compare the four methods described using these four characteristics in table 3.5.1, and the table follows naturally from the discussion of each of the four methods. Given these characteristics, which method should one choose to learn common factors across multiple datasets? There are no hard rules, but we propose first of all choosing a method whose solution is unique, so that we do not have the additional need to choose one solution among the possible solutions. In addition, for our purpose, we would like to be able to choose a specific number (e.g. K) of factors from all the factors learned by a particular method. This is straightforward when the factors' order of importance is shared across all the datasets so that we can just select the K most important factors. In contrast, it is not clear how to do this in the case of (HO)GSVD, because the K most important factors for one dataset might not be important in another dataset, and we might lose important information. As a consequence, for this thesis, in our experiments we consider PCA and CCA and we leave out GSVD and orthogonal FA for future investigation.

One main difference between PCA and CCA is in how each method treats the datasets, whether each dataset is explicitly modeled as being a separate entity or whether the datasets are concatenated together. Given the basic form of each method (equation (3.59) for PCA and equation (3.44) for CCA), it is not clear which method is preferable. However, with CCA we also have the ability to perform regularization while it is not clear how to do so in the context of PCA. This is especially important in scenarios where we are prone to over-fit the data, and working with high-dimensional data like the fMRI data certainly provides such a scenario. From this perspective, CCA should be more desirable compared to PCA. In addition, with CCA we also have the choice of using the formulation given in equation (3.56)—where we concatenate some of the datasets together—as opposed to the formulation of equation (3.44)—where we consider each dataset as being a totally separate entity. There does not seem to be a way to analytically prove that one of these two formulations is better than the other, so we perform empirical comparison instead using actual data in chapter 5.

3.6 Imputation when there are non-matching instances

All the methods we have considered so far assume that all the datasets have matching instances. This assumption might be too restrictive if we are going to apply any of these methods to analyze together fMRI data from different studies in general, because it is uncommon that we can match all instances or trials in different fMRI experiments. In this section, we present a way to relax the original assumption and have the multiple datasets share subsets of instances instead.

Let there be M datasets, and let $I_m \subseteq \{1, \cdots, N\}$ be the set of instances for dataset $m, 1 \leq m \leq M$. Here, the cardinality of $\cup_{m=1}^{M} I_m$ is N. As before, we represent each data m with an $D_m \times N$ matrix $\mathbf{X}_m$; however, now only a subset of the columns of $\mathbf{X}_m$ have actual values, corresponding to the instances I_m present in dataset m, and the values of the remaining columns are considered missing. The general approach we consider here is to find a way to fill the missing values so that we can apply any of the methods previously considered to analyze these datasets together. In statistics, this is referred to as the missing-value imputation problem.

Now we need to find a procedure to fill in the missing values in such a way that when the imputed matrices are fed to any factor analysis estimation methods we still get factors that have good predictive values. We consider a procedure adapted from the KNNimpute procedure presented in Troyanskaya et al. (2001) for the imputation of gene expression data. As is the case with the KNNimpute procedure, the procedure presented here is based on the notion of nearest neighbors. The idea is, given a particular instance missing in a particular feature of a particular dataset, we look at the closest features in the other datasets to obtain imputed value. However, unlike the KNNimpute procedure which is designed to handle missing values that can occur in arbitrary features and instances in one dataset, our procedure is meant to be applied to the case where we have multiple datasets and where in a specific dataset, the missing value cases apply to all features corresponding to a specific subset of instances. As a consequence, while in the original KNNimpute procedure each imputed value is derived from other values in the same dataset, in our case,

we need to derive the imputed values for a specific dataset from values present in some of the other datasets. The procedure is shown in algorithm 1, where we use MATLAB's notation to index vectors and matrices.

Algorithm 1 Impute a matrix entry using k-nearest-neighbors

Require: data matrices $\{\mathbf{X}_m\}_{m=1}^{M}$;
$\hat{m}$ the index of the dataset to impute;
$\hat{d}$ the feature (row) to impute;
$\hat{n}$ the instance (column) to impute;
K the number of nearest neighbors

1: initialize data structure to store distance information L
2: **for all** m such that $1 \leq m \leq M$ and the $\hat{n}$-th column is not missing in $\mathbf{X}_m$ **do**
3: $I_{\text{curr}} \leftarrow I_m \cap I_{\hat{m}}$
4: **for all** d such that $1 \leq d \leq D_m$ **do**
5: find dist($\mathbf{X}_{\hat{m}}(\hat{d}, I_{\text{curr}})$, $\mathbf{X}_m(d, I_{\text{curr}})$) and store the distance information in L
6: **end for**
7: **end for**
8: sort L based on ascending distances
9: obtain the K nearest features, $\mathbf{x'}_{(1)}, \cdots, \mathbf{x'}_{(K)}$ from L
10: totalDist $\leftarrow$ the total distances of $\mathbf{x'}_{(1)}, \cdots, \mathbf{x'}_{(K)}$
11: impute $\leftarrow 0$
12: **for all** k such that $1 \leq k \leq K$ **do**
13: $d \leftarrow$ dist($\mathbf{X}_{\hat{m}}(\hat{d}, :), \mathbf{x'}_{(k)}$)
14: $w \leftarrow$ (totalDist - d) / ((K - 1) * totalDist)
15: impute $\leftarrow$ impute + $w * \mathbf{x'}_{(k)}(\hat{n})$
16: **end for**
17: $\mathbf{X}_{\hat{m}}(\hat{d}, :) \leftarrow$ impute

The procedure computes an imputed value for $\mathbf{X}_{\hat{m}}(\hat{d}, \hat{n})$, i.e. the $\hat{d}$-th row and the $\hat{n}$-th column of the data matrix $\mathbf{X}_{\hat{m}}$. First, it computes the distance of the $\hat{d}$-th row of $\mathbf{X}_{\hat{m}}$ with all the rows of the other datasets whose $\hat{n}$-th instance is not missing. This is shown in the loops in lines 2-7. The distance is based on those instances which are not missing in both the dataset-of-interest $\hat{m}$ and the other dataset, referred to as the impute dataset. If Euclidean distance is used, the distance might depend on the number of shared instances, which can vary between different impute datasets. For this reason, the actual distance we use is the (squared) Euclidean distance normalized by the number of shared instances. After all the distances are computed, we sort them and find the K features (rows of the impute datasets) that are closest to $\mathbf{X}_{\hat{m}}(\hat{d}, :)$; these features are referred to as the impute features. We compute the imputed value, shown in lines 10-17 by taking a weighted average of the $\hat{n}$-th instances in the impute features, where the weight depends on each impute feature's distance. Closer features are given bigger weights.

If we run the above procedure on each missing entry on all the data matrices, we obtain imputed data matrices that can then be fed to any of the factor analysis estimation procedures described previously. In the next chapter, we show the results of experiments applying this imputation procedure on actual fMRI datasets.

The basic procedure shown in algorithm 1 might not be very efficient if we implement it as is by computing all distances. For this reason, the actual algorithm is implemented using kd-trees (Friedman et al. (1977)), which improve the efficiency of the distance calculations.

3.7 Summary

In this chapter, we have presented the general factor analysis model as a way to integrate fMRI data from multiple subjects and multiple studies. The model does not require that the data are normalized to a com-

mon space. We have also presented several ways to estimate the parameters of the model, with a focus on canonical correlation analysis. One condition of the factor analysis model is that it requires that all the datasets share matching instances. To make the model flexible, we have described an approach based on nearest neighbors that handles the case when the datasets might not have matching instances.

We mentioned the second way to obtain estimates of the orthogonal factor analysis is through a probabilistic formulation. Probabilistic formulations also exist for PCA (Tipping and Bishop (1999) and Roweis (1998)) and CCA (Bach and Jordan (2005)). One advantage given by probabilistic formulations is that they make it possible to consider missing data simultaneously with the estimation of the parameters, most commonly through the EM algorithm. Hence, these formulations would not have any problem dealing with non-matching instances. However, in this thesis we do not consider these probabilistic formulations and leave them for future work.

Before we conclude the chapter, let us revisit the first question posed at the beginning of the chapter: Can we integrate multiple fMRI datasets without the need for normalization to a common feature space? We have shown in this chapter that the linear factor analysis framework provides a way to integrate multiple datasets having different feature spaces. This framework can be applied to fMRI data, so the answer to this question is yes.

In the next chapter, we shall see how effectively the factor analysis framework presented in this chapter works when applied to real fMRI data.

Chapter 4

Case Study 1

Abstract

This chapter describes the first application of the linear factor analysis framework described in the previous chapter. We use PCA and CCA to learn factors common across the fMRI data, and use the resulting factors in the context of the task of predicting fMRI activations associated with concrete nouns. The experiments use data from two fMRI studies that study the brain activations associated with concrete objects: the Word-Picture (WP) and Word-Only (WO) studies.

There are two parts of the case study. In the first part, we apply the framework using all the instances in the two datasets; since both studies use the same 60 concrete nouns, we can find correspondence between one instance (corresponding to a particular concrete noun out of the 60 concrete nouns) in one study and an instance (corresponding to the same noun) in the other study, so in this case, both studies have matching instances. In this case, we see that using common factors learned by CCA, we can obtain significantly better accuracies compared to the baseline model that does not integrate information across subjects and/or studies. We verify that the improvements are due to the integration of information. The effect of combining data across studies is seen when we use a few factors: doing that, we see faster improvements in accuracies, i.e. we obtain accuracies closer to the peak accuracies with fewer number of factors, compared to when we consider within-study data only.

In the second part, we perform experiments taking out some instances out of the data from the WP study and a different set of instances out of the data from the WO study, and applying a method that imputes the missing instances in each dataset. The results show that the imputation method has potential especially when each dataset to be analyzed has a unique set of instances, i.e. when no two datasets have perfectly matching instances.

We are now ready to see how the various factor analysis approaches perform on real fMRI data. In the next two chapters, we describe two case studies of the application of these approaches on the WP and WO datasets described in sections 1.3.3 and 1.3.4. In both case studies, the predictive task is to predict fMRI activations for unseen concrete objects.

In the first case study, described in this chapter, we investigate learning common factors from fMRI data across multiple subjects and multiple studies using some of the factor analysis methods described in chapter 3, and how effective the learned factors are when used in conjunction with a predictive model of fMRI activations for unseen objects. In the second case study, described in chapter 5, we investigate incorporating additional non-fMRI datasets, in particular, datasets obtained from large text corpus and behavioral study, in learning the common factors and the efficacy of the resulting learned factors in the same predictive setting as that used in the first case study.

4.1 Description

In the first case study, we investigate whether by integrating the fMRI data across subjects and potentially across studies using the linear factor analysis framework described in the previous chapter, we can obtain better predictive accuracy compared to a model without any integration of cross-subject cross-study fMRI data. In particular, in this case study, we consider the problem of predicting fMRI activations associated with the meaning of concrete nouns.

As described in section 2.3.1, Mitchell et al. (2008) propose a computational model associated with the meaning of concrete nouns, shown again in figure 4.1.1. The model posits that the fMRI activations associated with the meaning of concrete nouns can be characterized by some predefined semantic features, denoted as *base features* in the figure. In particular, Mitchell et al. (2008) use as semantic features the co-occurrence counts of the concrete nouns with a set of 25 verbs as derived from some large text corpus data. The semantic features are mapped to each voxel's activations linearly, and the mappings are learned using multivariate linear regression. Mitchell et al. (2008) show that this model can be used to predict the fMRI activations associated with novel concrete nouns with better-than-random predictive accuracies.

As shown in figure 4.1.1, in the baseline model of Mitchell et al. (2008), there is a separate mapping from the base features to the fMRI activations of a particular subject in a particular study, and each mapping is learned separately without any consideration about the mappings for other subjects. In this case study, we ask the question whether we can improve the predictive accuracies by integrating information present in the fMRI data across multiple subjects and/or studies. In particular, we augment the baseline model of 4.1.1 using our linear factor analysis framework.

The augmented model is shown in figure 4.1.2, and is referred to as the *fMRI-common-feature* model. Like in the baseline model, we assume that in the fMRI-common-factor model, there is a set of base features underlying the fMRI activations. However, unlike in the baseline model where the base features directly map to fMRI activations, in the fMRI-common-feature model, the base features map linearly to a set of features common to the fMRI data across subjects and studies, denoted as *learned common features* in the figure, and the common features then map linearly to the fMRI activations for each subject in each study. The model in turn is estimated in two stages. First, we learn the common features. These features are learned using the fMRI data from all the available subjects in all the available studies. In particular, we cast the learning of the common features as the problem of learning the factors in the linear factor analysis framework described in chapter 3. So we can use any of the methods described in that chapter to obtain the learned common features. In the second stage, we learn the mapping from the base features to the learned common features using multivariate linear regression.

In this case study, we perform two groups of experiments using the WP and WO datasets described in sections 1.3.3 and 1.3.4. The aim of the first group of experiments is to test whether we can obtain better predictive accuracies using the fMRI-common-feature model compared to the baseline model on these two datasets. In the second group of experiments, we investigate how effective the imputation scheme described in section 3.6 is when we try to integrate data from the two datasets in the case some of the words are not present in either dataset.

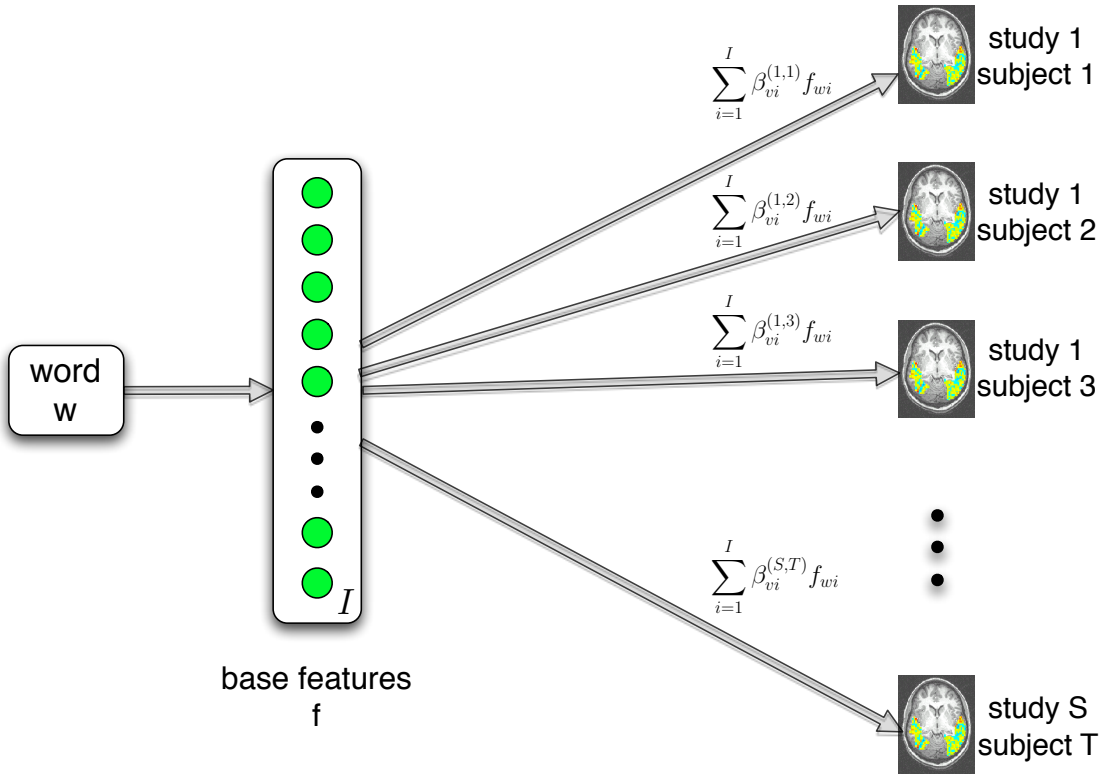


Figure 4.1.1: The baseline model: the predictive model of Mitchell et al. (2008), expanded to take into account the potential presence of fMRI data from multiple subjects and/or studies, with semantic features denoted as base features.

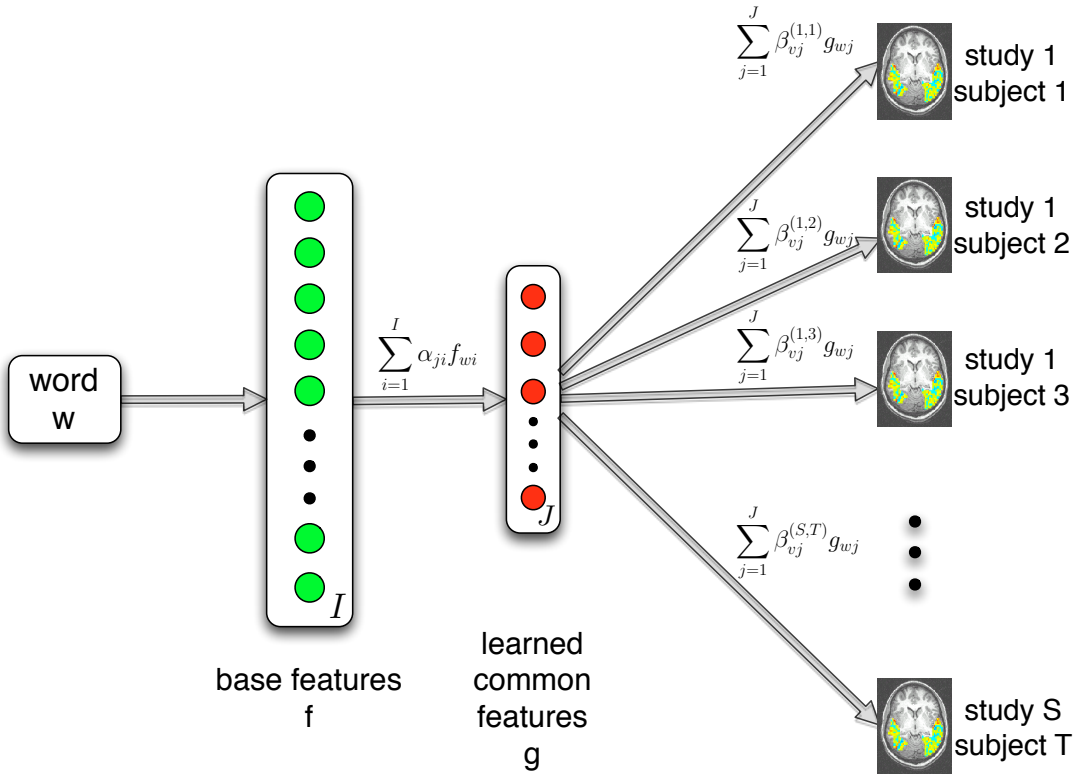


Figure 4.1.2: The fMRI-common-feature model, augmenting on the model of Mitchell et al. (2008)

4.2 Experiments: all instances

Here are the questions that we try to answer with the first group of experiments in case study 1:

1. Do we obtain better accuracies in the fMRI-common-feature model compared to the accuracies of the baseline model?
2. If the answer to the previous question is yes, is the improvement in accuracies due to the integration of fMRI data across subjects and/or studies?
3. What is the number of learned common features that yields the best accuracies?
4. What effect does integration across studies have, both in terms of the learned common features and in terms of accuracies?
5. How is each learned common feature reflected on the brain for each subject?
6. What kind of semantic information, if any, is present on the learned common features?
7. How many subjects are needed in order to obtain reliable learned common features, as measured by the resulting accuracies?

4.2.1 fMRI Datasets

The WP and WO datasets described in sections 1.3.3 and 1.3.4 are used. We refer to these sections for descriptions about how the data were pre-processed.

4.2.2 Predefined semantic features

In the experiments, we use two kinds of predefined semantic features as base features, the first kind derived from large text corpus data and the other derived from behavioral data.

485verb The way different words are used or not used together is closely related to what each of the words mean. Using this idea, Mitchell et al. (2008) used as predefined semantic features the co-occurrence counts of the 60 stimulus words used in the WP study with a set of reference words consisting of 25 handpicked verbs. In other words, in their experiments, there are 25 features, one for each verb in the reference set, and the value of each feature j for a stimulus word w is the number of times the word w co-occurs with the corresponding verb j. Mitchell et al. (2008) obtained these co-occurrence counts from a corpus of English web pages provided by Google, in particular by determining whether the word pair of interest occur together within five tokens of one another.

In our experiments, we also use co-occurrence counts as a kind of predefined semantic features. One difference from the work described in Mitchell et al. (2008) is that we use a bigger set of reference words, containing 485 verbs. This set also contains the 25 verbs used by Mitchell et al. (2008). The co-occurrence counts are again derived from the Google corpus used to derived the counts used in Mitchell et al. (2008). These features are referred to as the 485verb features.

intel218 Another characterization of the semantic information present in words, in particular for concrete nouns as used in the WP and WO studies, can be provided by answers to questions about properties of the objects corresponding to the words. Given this idea, we use as another kind of predefined semantic features answers to 218 yes/no questions about various objects gathered by Dean Pomerleau at Intel Research Pittsburgh (Palatucci et al. (2010)). Note that these features are behavioral in nature, since they are based on responses gathered from humans. These features are referred to as the intel218 features.

Pre-processing for the predefined semantic features Raw co-occurrence counts can take values in the order of thousands or more, while fMRI activations typically have magnitude of less than one. In order to

match the scaling of the co-occurrence counts with that of the fMRI activations, we normalize the 485verb features so that for each instance (stimulus word) it has length one.

4.2.3 Evaluation

Obtaining an accuracy estimate We use the evaluation procedure described in section 2.3.1.2 to evaluate the predictive performance of each of the approaches. As mentioned in that section, it is also desired to obtain a confidence interval for each accuracy estimate. A procedure to do this is outlined next. Because of the computationally intensive nature of the confidence interval estimation procedure, we have confidence interval estimates only for the results of case study 2 in the next chapter.

Estimating the confidence interval for an accuracy estimate Because of the procedure used to compute the accuracy, it is not clear that we can resort to the standard parametric ways to construct a confidence interval. It is also complicated by the fact that the folds are not necessarily independent, since a particular word can be left out in multiple folds. To address this, we construct the confidence interval using a nonparametric approach. In particular, we use the jackknife approach[1].

The *jackknife* approach (Efron (1982)) is a nonparametric approach to compute the distribution of a particular statistic of the data. The data is assumed to consist of N independent instances. The jackknife works by leaving out one instance of the data and computing the statistic based on the remaining instances, and repeating the procedure until each instance has been left out exactly once. In some sense, this is similar to the leave-one-out cross-validation procedure. In the end, the procedure produces N values for the statistic of interest, which gives an indication of the statistic's distribution. These values can then be used to quantify the uncertainty of the statistic, such as standard deviation or confidence interval.

Now we see how the jackknife can be used to obtain the uncertainty of the accuracy estimate obtained using the procedure described in section 2.3.1.2. We mentioned above that each fold is not necessarily independent of the other folds. We can, however, assume that the data associated with a word out of the sixty words used in the WP and WO experiments are independent of the data associated with the other words. Based on this assumption, we can apply the jackknife by leaving out a particular word and applying the cross-validation procedure described in section 2.3.1.2 on the remaining 59 words. When we do this, we obtain the leave-two-word-out cross-validation accuracy on the 59 remaining words. We can repeat this by taking out a different word out of the 60 words to compute another 59-word cross-validation accuracy, and when the procedure is repeated 60 times—with each of the 60 words taken out in one out of the 60 repetitions—we obtain 60 accuracy numbers. We can then used these accuracy numbers to obtain a 95% jackknife confidence interval for the original accuracy by taking the 2.5- and 97.5-percentiles of the 60 accuracy numbers.

What makes the jackknife approach valid? We can compute the distribution of a statistic of the data by obtaining several samples of the data and computing the statistic for each of the data sample. However, in most cases, which is also mostly the case for fMRI data, we have only one or very few samples of the data, and obtaining additional samples of the data is not practical. In this situation, one helpful insight is that given the sample instances are independent, subsets of the sample can be considered as additional samples of the data. In particular, based on this insight, each jackknife repetition, which is applied to a subset of the original sample, effectively works on a different sample of the data. An alternative approach is to obtain additional samples by uniformly sampling from the original sample with replacement. Intuitively, this approach bootstraps the data generation process using the available sample, and for this reason, the approach is referred to as the *bootstrap* approach (Efron (1982)). Given these two approaches—the jackknife and the bootstrap—is one superior to the other, and which one should we choose? In essence, there are additional uncertainties associated with both approaches: both produce an approximation to the true distribution of the statistic of interest. Ideally we use both approaches so that we get a better view of the uncertainty involved, but because of computational reasons, in this thesis we use only the jackknife approach.

[1]This approach was suggested to us by Jay Kadane.

Based on the duality between confidence intervals and hypothesis tests (see, for instance, theorem 9.2.2 in Casella and Berger (2002)), the confidence intervals obtained using the jackknife approach can be used to test whether one accuracy estimate is statistically significantly different from another accuracy estimate. We do this by looking at the confidence intervals for the two accuracies. If the confidence intervals do not overlap, we declare that the accuracies are significantly different. If the intervals overlap, however, it is still possible that the accuracies are significantly different, but we do not have enough evidence to declare this with sufficient certainty. In the latter case, we admit the reasonable possibility that the two models being compared are equivalent in terms of predictive performance. In a sense, this is a conservative test.

4.2.4 Methods

In the first group of experiments, we try the following methods:

- **LR**, the baseline model of Mitchell et al. (2008)
- **PCA-concat**, an instantiation of the fMRI-common-feature model where we learn common features for the data coming from subjects within only a particular study, and with the common features learned by concatenating the all the subjects' data matrices and applying principal components analysis (PCA) to the concatenated matrix; **PCA-concat-WP** and **PCA-concat-WO** refer to specific instances of this method when applied to data from the WP and WO studies, respectively
- **PCA-concat-comb**, an instantiation of the fMRI-common-feature model where we learn common features for the data coming from subjects from across the two studies, and with the common features learned by concatenating all the subjects' data matrices and applying PCA to the concatenated matrix
- **CCA-mult**, an instantiation of the fMRI-common-feature model where we learn common features for the data coming from subjects within only a particular study, and with the common features learned by applying canonical correlation analysis (CCA) to all the subjects' data matrices; **CCA-mult-WP** and **CCA-mult-WO** refer to specific instances of this method when applied to data from the WP and WO studies, respectively
- **CCA-mult-comb**, an instantiation of the fMRI-common-feature model where we learn common features for the data coming from subjects from across the two studies, and with the common features learned by applying CCA to all the subjects' data matrices

In addition to integrating fMRI data across subjects and studies, the fMRI-common-feature model also does dimensionality reduction. One might ask which of these two aspects—cross-subject-study fMRI data integration or dimensionality reduction—plays a bigger role in the fMRI-common-feature. To try to address this question, we also use a variation of the fMRI-common-feature that involves only dimensionality reduction, called the *fMRI-dimensionality-reduction* model and shown in figure 4.2.1.

We see that like in the fMRI-common-feature model, the fMRI-dimensionality-reduction also inserts some learned features between the base features and the fMRI activations. However, unlike in the fMRI-common-feature model, in the fMRI-dimensionality-reduction, there is a different set of learned features for each particular subject and study's fMRI activations, and these features are learned based on only the corresponding subject and study's fMRI activations. We have two methods that are instantiations of the fMRI-dimensionality-reduction model:

- **PCA-indiv**, use PCA to learn the features
- **CCA-indiv**, use CCA to learn the features; the way this is done is, since CCA needs at least two data matrices, for a particular subject, we apply CCA to two copies of that subject's data matrix; we try this method in order to have an instantiation of the fMRI-dimensionality-reduction model that possesses the constraints of CCA

One caveat with the CCA-indiv method is that it is ill-posed[2]. This is because as long as the loadings for the two copies of the dataset are the same, the correlation between the projections will always be one. As a result, from the perspective of the canonical correlation analysis method, any loading values that are

[2]Thanks to Zoubin Ghahramani for pointing this out.

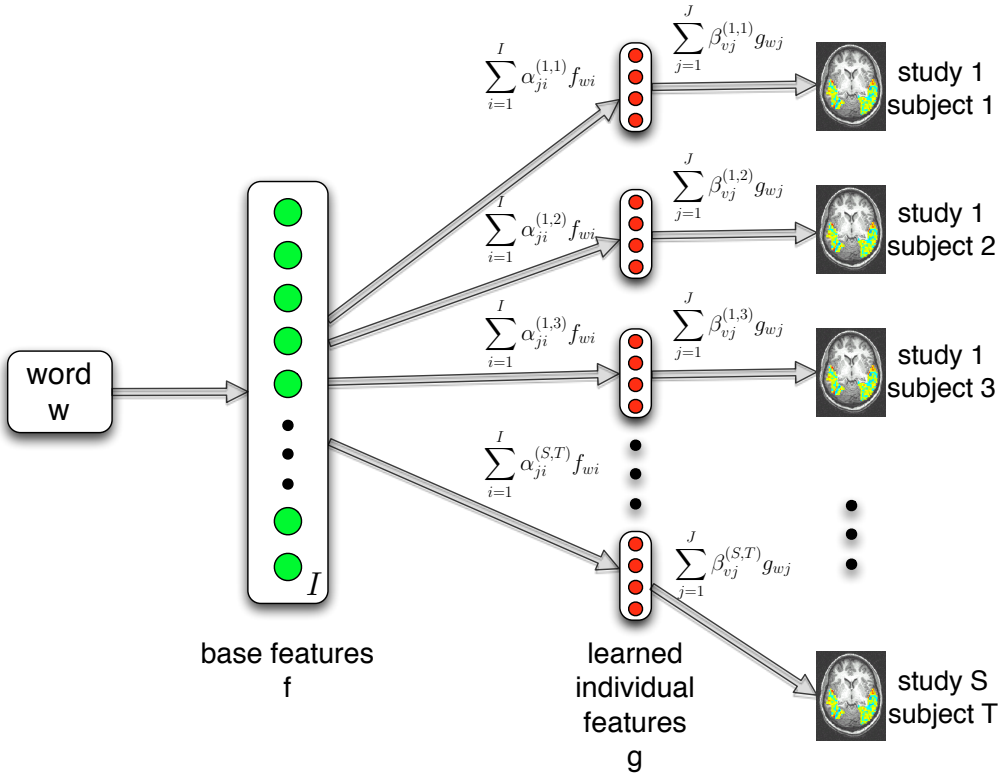


Figure 4.2.1: The fMRI-dimensionality-reduction model, another variation of the model of Mitchell et al. (2008)

the same for the two copies will be a solution. We should therefore take this into account when considering the results of the CCA-indiv method. For the CCA-indiv results reported here, we use the solutions as generated by the LAPACK[3] function `dsygv` for solving a symmetric generalized eigenvalue problem.

For methods that perform dimensionality reduction (instantiations of both the fMRI-common-feature and the fMRI-dimensionality-reduction models), we try different numbers of features (1, 2, 3, 4, 5, 10, 20, 30, 40, 50 features). For the methods that use CCA, we use the kernel implementation of CCA with regularization, setting $\kappa = 0.5$ as the regularization parameter. We do not explore nor make any claims that this is the optimal regularization parameter to use.

4.2.5 Results

Accuracies The results of the first group of experiments are summarized in figure 4.2.2:

- top-left: the mean accuracies across the WP subjects for the various methods when used in conjunction with the 485verb features
- top-right: the mean accuracies across the WO subjects for the various methods when used in conjunction with the 485verb features
- bottom-left: the mean accuracies across the WP subjects for the various methods when used in conjunction with the intel218 features
- bottom-right: the mean accuracies across the WO subjects for the various methods when used in conjunction with the intel218 features

[3] `http://www.netlib.org/lapack/`

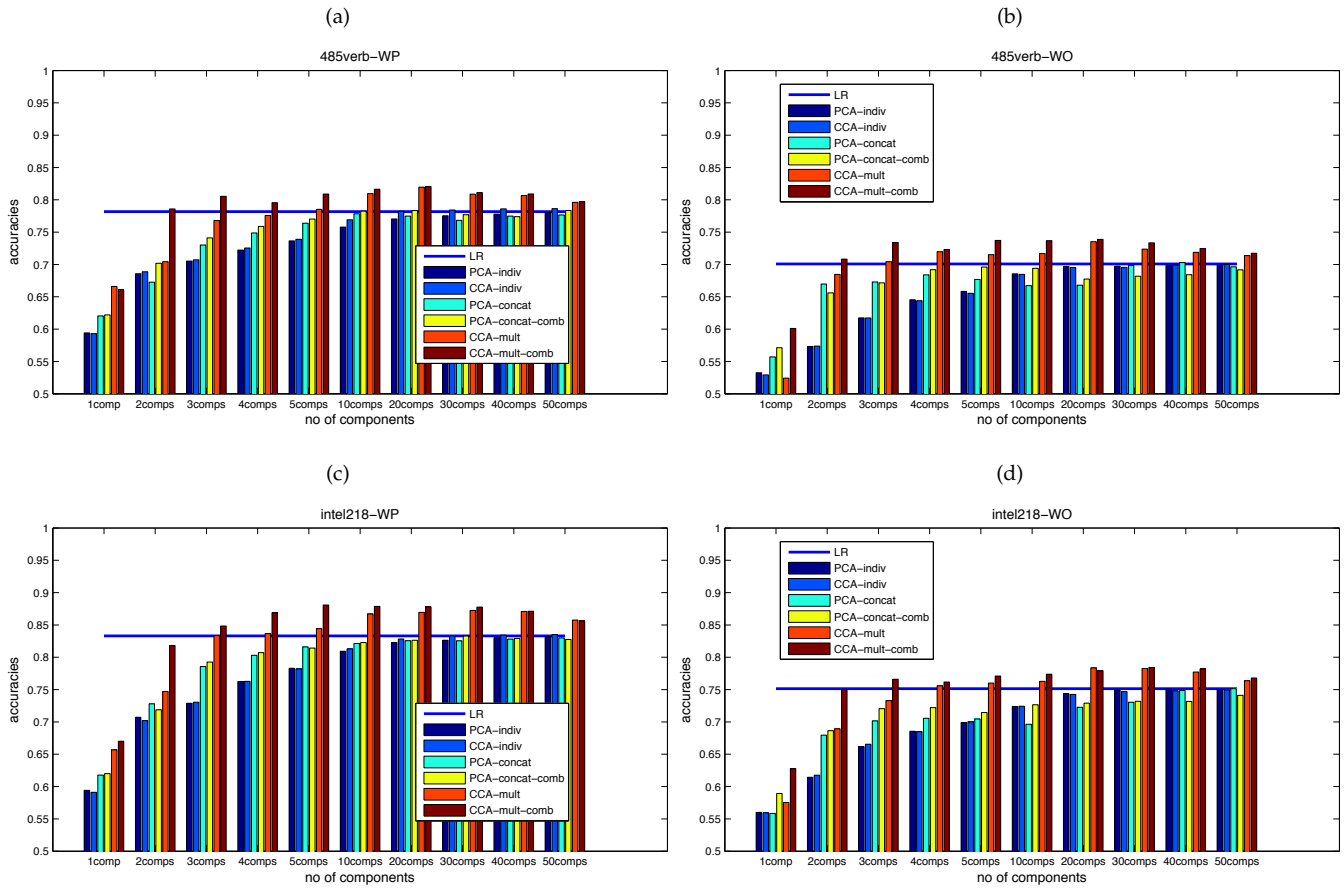


Figure 4.2.2: Accuracies, averaged over the subjects in each study (WP and WO), for the baseline method of Mitchell et al. (2008) (LR, blue horizontal line) and the PCA-indiv, CCA-indiv, PCA-concat, PCA-concat-comb, CCA-mult, CCA-mult-comb methods. The first row show the accuracies when using the 485verb features, and the second row show the accuracies when using the intel218 features.

In all four cases, we see that both the CCA-mult and the CCA-mult-comb yield significantly better accuracies compared to the baseline model when sufficient number of learned features are used. In particular, the highest accuracies for these methods are obtained when 20 learned features are used. At the highest accuracies, there are only marginal differences between the accuracies of the CCA-mult and the CCA-mult-comb methods across the four scenarios. So for the numbers of components yielding close to the optimal accuracies, integrating across studies does not give us significant improvements compared to integrating within a particular study. However, when we use significantly fewer learned features, as we increase the number of learned features, for the CCA-mult-comb method we get accuracies closer to the peak accuracies with fewer numbers of learned features compared to the numbers of learned features needed for the CCA-mult method to achieve similar levels of accuracies. This indicates that by integrating across studies, the CCA-mult-comb method is able to put more information relevant for prediction in the first few learned features.

On the other hand, the same trends are not present for the PCA-concat and the PCA-concat-comb methods. For these methods, instead we see that their accuracies are never significantly better compared to the baseline model's accuracies, although we do see the trend that the accuracies go up as the number of learned features increases, and when 50 features are learned , the accuracies are comparable to the baseline model's accuracies. This seems to suggest that CCA integrates the cross-subject cross-study fMRI data

more effectively compared to PCA. Note that CCA explicitly models the different datasets as being separate entities, while in PCA the different datasets are combined together into a single dataset. This might play a role in the difference in the effectiveness of each method in integrating the different datasets. We also note that we use regularization for CCA, while the multivariate formulation of PCA does not suggest any straightforward way to do the equivalent regularization, especially since there is already a constraint on the norm of the projection vectors in PCA (equation (3.5)). We investigate this more a little bit later.

Furthermore, the methods that are instantiations of the fMRI-dimensionality-reduction model, i.e. the PCA-indiv and CCA-indiv methods, also do not significantly outperform the baseline model regardless of the number of features used, although in this case we again see that their accuracies improve as the number of features increases, roughly matching the accuracies of the baseline model when there are at least 40 learned features. The results suggest that the improvements that we see with the CCA-mult and CCA-mult-comb methods are due to the integration of cross-subject (and potentially cross-study) fMRI data instead of dimensionality reduction.

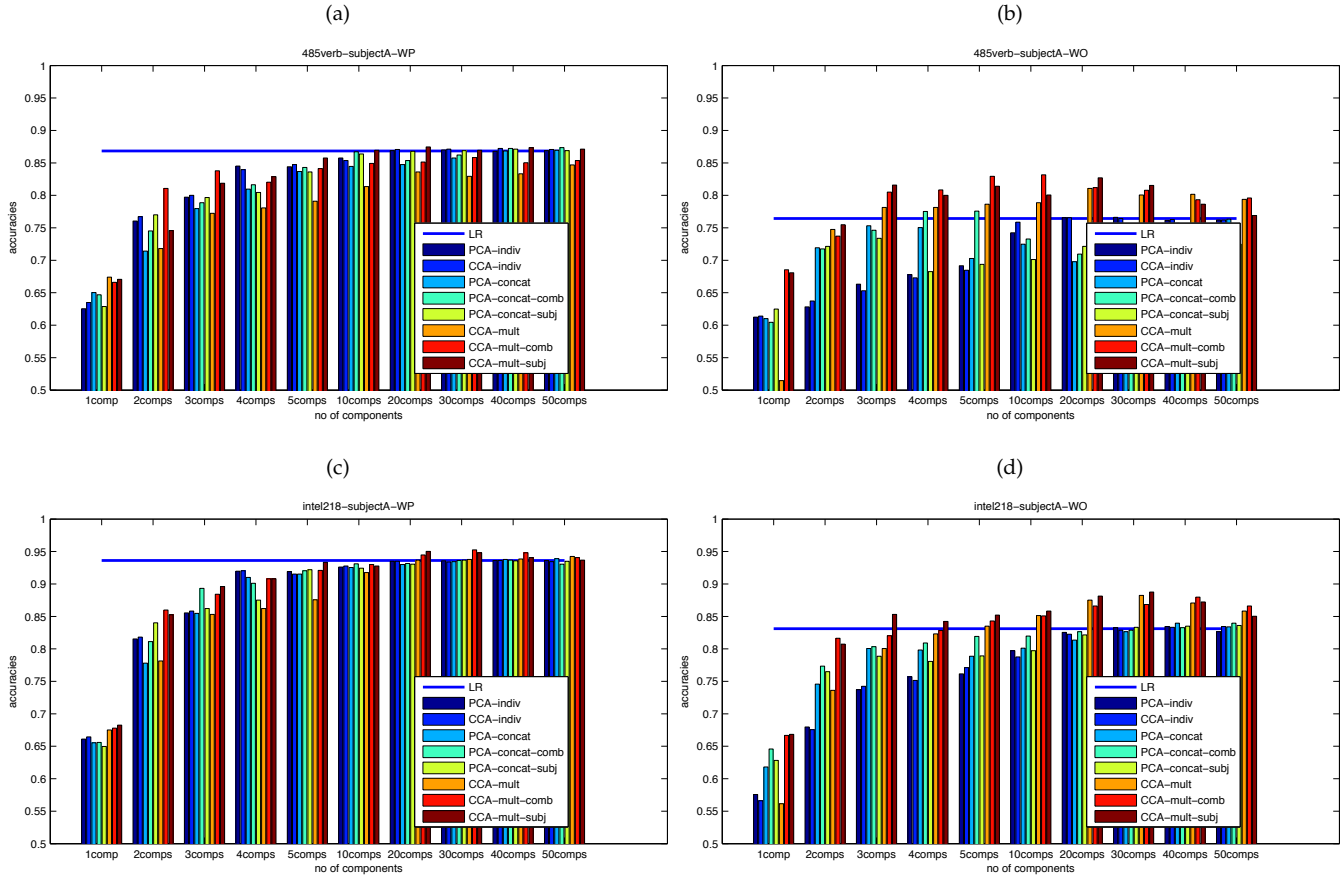


Figure 4.2.3: Individual accuracies for the common subject A.

We now investigate how the accuracies vary for different subjects. In particular, we focus on the three subjects that participated in both studies, labeled as subjects A, B, and C. These accuracies are shown in figures 4.2.3, 4.2.4, and 4.2.5, for subjects A, B, and C respectively. In these figures, we also the accuracies for two additional methods: **PCA-concat-subj** and **CCA-mult-subj**. These are methods where we integrate fMRI data of only the respective subjects from both studies, e.g. in the subject A case, both the PCA-concat-subj and the CCA-mult-subj methods learn common features using only subject A's fMRI data from the two studies WP and WO, and similarly for subjects B and C.

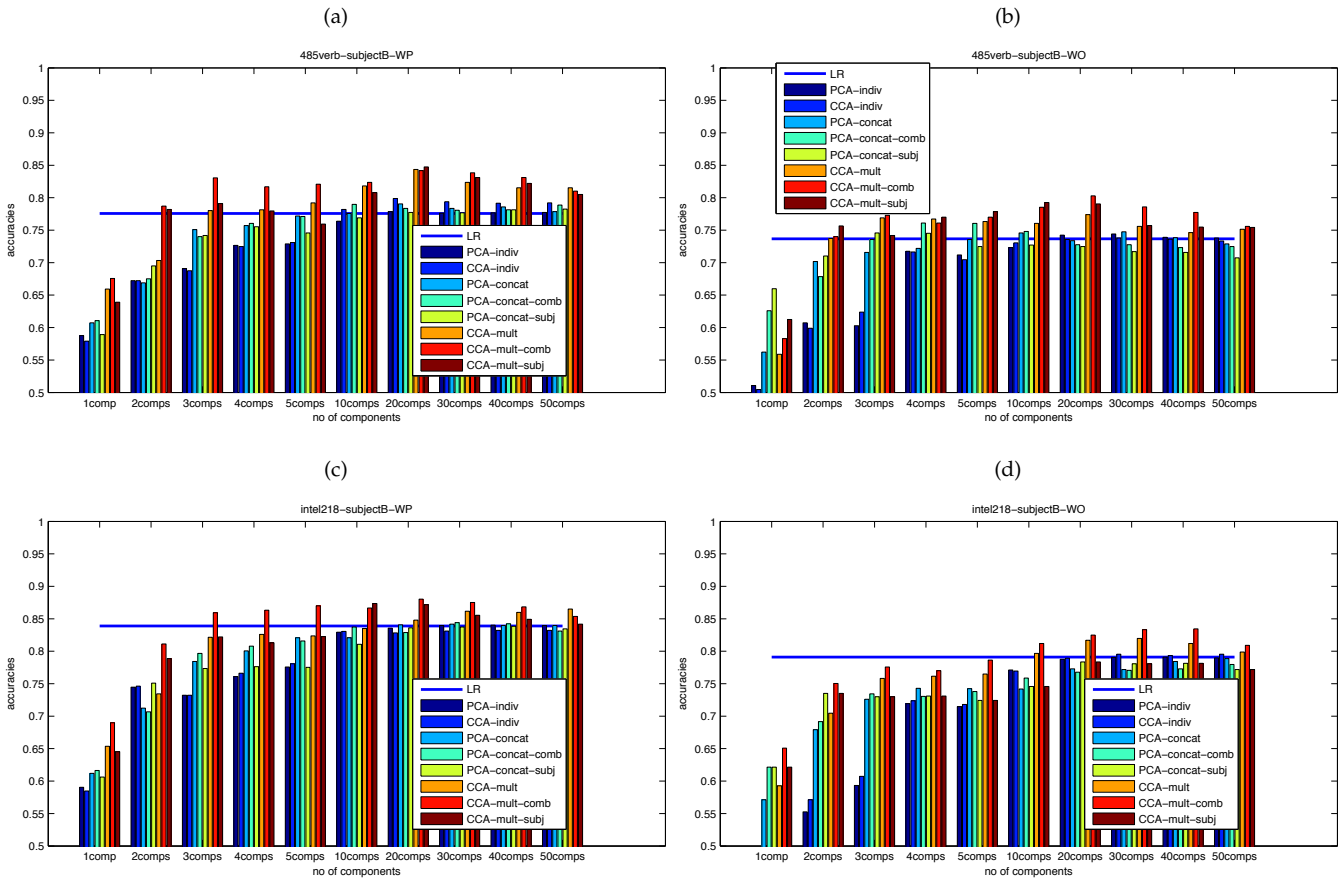


Figure 4.2.4: Individual accuracies for the common subject B.

In general, we see that trends appearing for the mean accuracies shown in figure 4.2.2 are also present for the three subjects' accuracies. An exception is the accuracies for subject A-WP, where none of the factor-based methods significantly outperform the baseline model's accuracies. However, note also that the subject A-WP case is one where the accuracies are high compared to the other cases.

In addition, with few exceptions, we see also that, like in the case for the counterpart methods CCA-mult and CCA-mult-comb, the accuracies for the CCA-mult-subj method can significantly exceed the baseline model's accuracies given sufficient learned features. This indicates that even when there are only two datasets to analyze jointly—in this case, each dataset coming from the same subject but from two different studies—the fMRI-common-feature can still integrate the information available in the two datasets effectively.

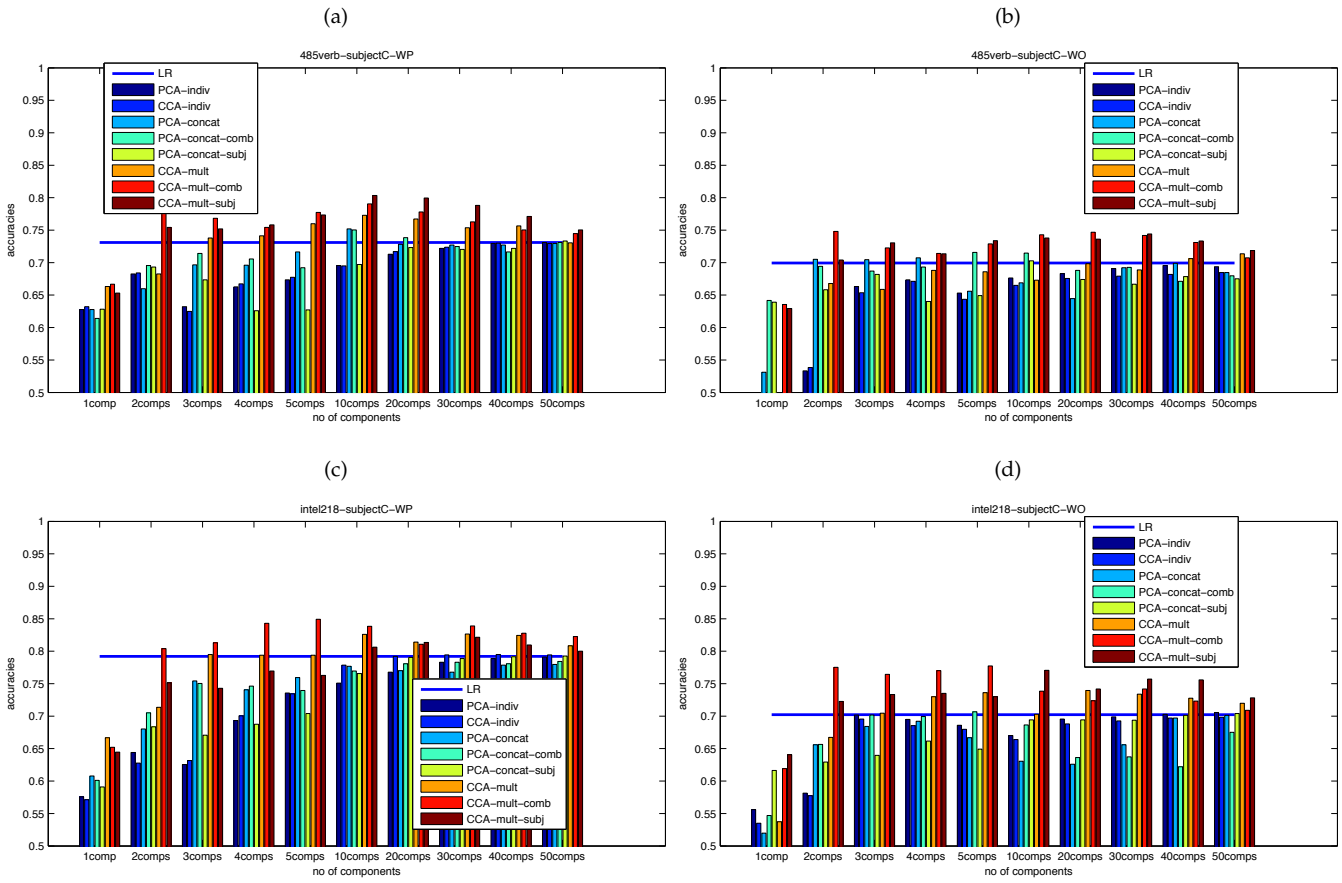


Figure 4.2.5: Individual accuracies for the common subject C.

Information contained in the learned features Now we look into the kind of information present in the learned features for methods that are instantiations of the fMRI-common-feature model. To do this, we can first look at the values assigned to each of the 60 words for a particular learned feature, based on learning the model using all 60 stimulus words. However, instead of looking at the actual values, we look at the relationship between the learned feature value for a particular stimulus word and the values for the other stimulus words. In particular, for each learned feature, we rank the stimulus words based on the learned feature values.

	component 1	component 2	component 3	component 4	component 5
positive	knife	carrot	horse	dog	corn
	screwdriver	spoon	cat	coat	chisel
	spoon	cow	fly	telephone	cup
	pliers	dog	bee	shirt	dresser
	hammer	screwdriver	beetle	refrigerator	key
	cat	bear	bicycle	bear	tomato
	key	pliers	dog	skirt	lettuce
	bottle	beetle	hand	apartment	table
	skirt	bed	bear	tomato	horse
	arm	arm	eye	beetle	arch
negative	apartment	leg	screwdriver	truck	cat
	barn	key	knife	car	closet
	church	pants	saw	train	arm
	house	chair	window	carrot	dog
	closet	glass	bottle	igloo	church
	window	chisel	spoon	corn	pants
	train	bottle	door	foot	igloo
	desk	cat	refrigerator	key	dress
	dresser	dress	chimney	desk	airplane
	arch	corn	glass	arm	chair

(a) CCA-mult-WP

	component 1	component 2	component 3	component 4	component 5
positive	screwdriver	celery	pliers	cup	screwdriver
	refrigerator	carrot	hand	corn	arm
	butterfly	lettuce	hammer	barn	refrigerator
	telephone	pliers	spoon	house	eye
	apartment	saw	key	carrot	fly
	bicycle	screwdriver	foot	church	dresser
	dresser	chisel	arm	igloo	corn
	lettuce	tomato	arch	celery	saw
	airplane	spoon	screwdriver	bell	cow
	bottle	knife	dresser	cow	bottle
negative	eye	car	cat	hand	train
	cat	apartment	cow	foot	carrot
	ant	house	dog	arm	truck
	car	desk	butterfly	leg	spoon
	cow	church	telephone	bicycle	watch
	leg	closet	refrigerator	horse	hand
	arm	bed	bee	butterfly	airplane
	fly	table	horse	pants	beetle
	bee	door	fly	dog	bear
	key	arch	airplane	airplane	chisel

(b) CCA-mult-WO

	component 1	component 2	component 3	component 4	component 5
positive	apartment	cat	leg	pants	corn
	church	car	key	dress	igloo
	closet	eye	foot	glass	key
	house	dog	arm	coat	cup
	barn	fly	chair	chair	eye
	window	cow	desk	skirt	bottle
	dresser	bee	hand	cat	tomato
	desk	ant	door	bottle	barn
	train	horse	pants	refrigerator	bell
	chimney	bear	arch	shirt	lettuce
negative	knife	screwdriver	telephone	car	arm
	cat	pliers	butterfly	spoon	foot
	spoon	refrigerator	bicycle	screwdriver	hand
	key	knife	beetle	saw	horse
	pliers	hammer	dog	carrot	airplane
	screwdriver	celery	bear	cow	leg
	chisel	bottle	refrigerator	house	bicycle
	saw	chisel	lettuce	truck	screwdriver
	fly	glass	apartment	pliers	butterfly
	hand	spoon	horse	knife	truck

(c) CCA-mult-comb

Table 4.2.1: The rankings of the the stimulus words in the first five components for the CCA-mult-WP (top left), CCA-mult-WO (top right), and CCA-mult-comb (bottom) methods.

Table 4.2.1 shows the rankings of the stimulus words for the first five componentsfeatures learned by the CCA-mult and CCA-mult-comb methods. For each component, we see groupings of ten words each labeled as *positive* and *negative*. The positive (negative) grouping refers to the top ten words (out of the 60 words used in the WP and WO studies) with the most positive (negative) scores for the respective component. In other words, for a particular component, if we do a descending (ascending) sort of the words based on the score of each word for that component and take the first ten sorted words, we obtain the positive (negative) grouping for that component. The kind of words included in the groupings for a particular component give an idea about the semantic dimensions that this component represents. Also note that we can interchange the labels of the groupings for a particular component, since the general linear factor analysis framework shown in equation (3.2) is invariant to changing the signs of both the loadings and the scores for each factor/component.

From the table, we see some interesting semantic dimensions being extracted by these methods. The first learned feature for the CCA-mult-WP method contains a grouping of tool words on one end of the spectrum and a grouping of shelter words on the other end of the spectrum. These groupings are also

Shelter	Manipulation	Eating	Word length
apartment	pliers	carrot	butterfly
church	saw	lettuce	screwdriver
train	screwdriver	tomato	telephone
house	hammer	celery	refrigerator
airplane	key	cow	bicycle
key	knife	saw	apartment
truck	bicycle	corn	dresser
door	chisel	bee	lettuce
car	spoon	glass	chimney
closet	arm	cup	airplane

Table 4.2.2: The common factors discovered by Just et al. (2010)

present to some extent in the first learned feature for the CCA-mult-comb method–here we see that the tool (shelter) grouping is labeled as positive (negative) for CCA-mult-WP, while the label is reversed for CCA-mult-comb, but this is not a problem given the invariance of each factor with respect to changing signs of both the loadings and the scores as noted in the previous paragraph. On the other hand, the first learned feature for the CCA-mult-WO method seems to group words based on the length of each of the words, so it has short words on one end and long words on the other end. The short-word dimension seems to also be present in the second learned feature for the CCA-mult-comb method. On the other hand, the tool-shelter dimensions present in the first learned feature for the CCA-mult-WP and CCA-mult-comb methods seem to be present to some extent in the second learned feature for the CCA-mult-WO method. Other semantic dimensions that are present include body-part (component 4 for CCA-mult-WO or component 5 for CCA-mult-comb) and apparel (4th feature CCA-mult-comb).

The results we just describe are similar to what is presented in Just et al. (2010). Like what is described here, Just et al. (2010) try to find common factors across subjects, in particular, across the subjects participating in the WO study. However, they find these factors by first finding subject-specific factors based on the 50 most stable voxels in each of the five lobes of the brain using the orthogonal factor analysis with varimax as described in chapter 3, and then aggregating the subject-specific factors for all the subjects and again applying factor analysis to these aggregated factors. In essence, they perform two-level factor analysis. To summarize, here are the main differences between the experiments described in this chapter and the experiments performed by Just et al. (2010):

1. Just et al. (2010) use the 50 most stable voxels in each of the five lobes (250 voxels total), while we use the 500 most stable voxels across the whole brain
2. Just et al. (2010) use two-level factor analysis to find the common factors, while we find factors in one step, using either PCA or CCA

Despite these differences, there are some commonalities between what we describe in this chapter and what Just et al. (2010) find. Just et al. (2010) find four factors common across the subjects, shown in table 4.2.2 along with the associated words, i.e. words with the highest factor scores for each factor. When we compare the factors and the words in table 4.2.2 with what are shown in table 4.2.1, we see some similarities. For instance, as mentioned above, we see groupings of shelter words in table 4.2.1, and the groupings of tools words we find are similar to the manipulation factor of Just et al. (2010). We also see groupings related to word-length, but only when we analyze the WO dataset on its own.

We can also see how each of the learned feature projects to the brain of each subject. To do this, we perform multivariate linear regression where we have the learned feature values as covariates and the fMRI activations as response variables. The resulting regression coefficients or loadings for four subjects (two WP subjects, 1WP and 5WP, and two WO subjects, 1WO and 11WO) and the first two learned features are shown in four figures:

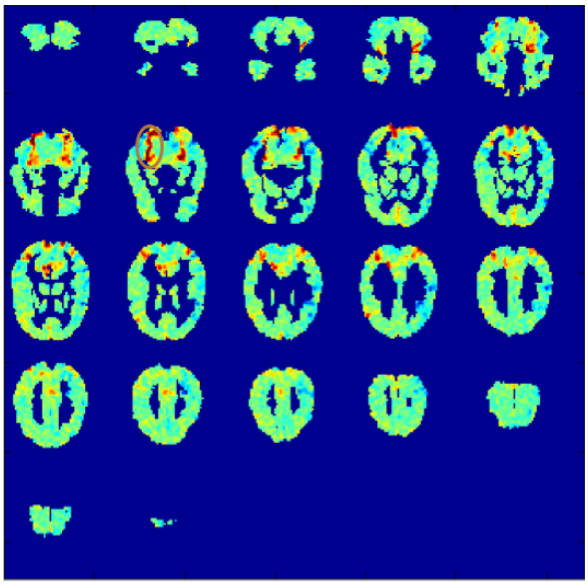
- figure 4.2.6

- figure 4.2.7
- figure 4.2.8
- figure 4.2.9

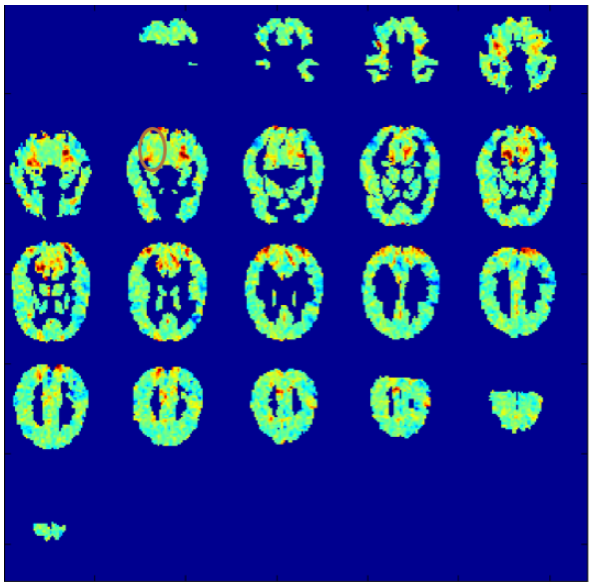
Note that subjects 1WP and 1WO correspond to subject A. Also, in these figures, and in subsequent figures where we show slices of the whole brain, going from left-to-right from the top left to the bottom right we go from the inferior part of the brain to the superior part, and in each slice, the top (bottom) part denotes the posterior/back (anterior/front) part of the brain while the left (right) part of the slice denotes the right (left) part of the brain.

First note that previously we mentioned that the shelter-tool dimension seems to be present in component 1 for the CCA-mult-WP and CCA-mult-comb methods, while being present in component 2 for the CCA-mult-WO method. We see this reflected in the loadings. The loadings for component 1 for the CCA-mult-WP method are highly similar to the corresponding loadings for component 1 for the CCA-mult-comb method, while the loadings for loadings for component 2 for the CCA-mult-WO method are highly similar to the corresponding loadings for component 1 for the CCA-mult-comb method. We see a strong projection to the fusiform areas corresponding to the shelter dimension for the WP subjects (component 1 for the CCA-mult-WP and CCA-mult-comb methods). This projection is present to a lesser extent for the WO subjects.

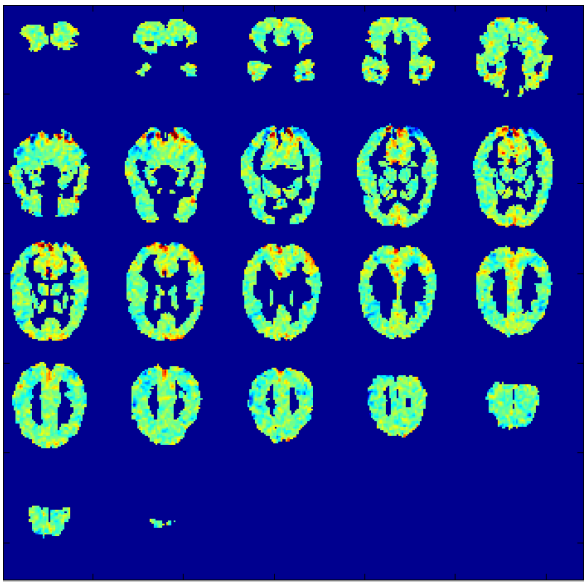
(a) subject 1WP, component 1



(b) subject 5WP, component 1



(c) subject 1WP, component 2



(d) subject 5WP, component 2

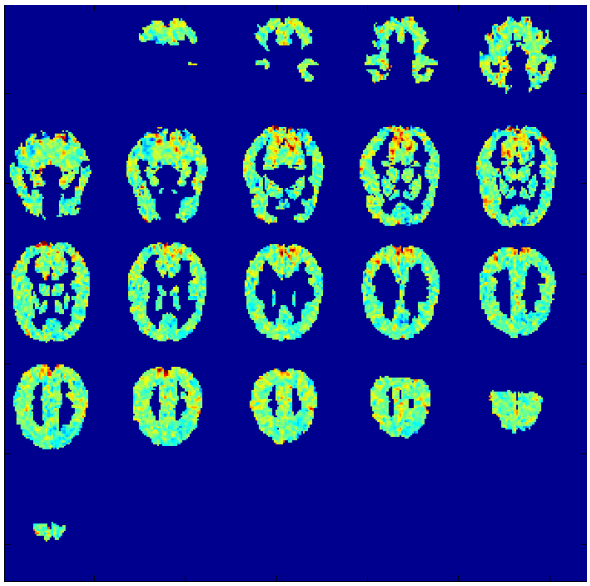
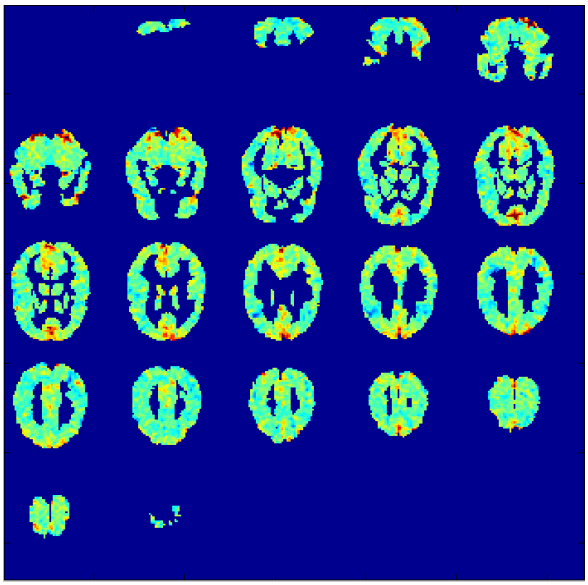
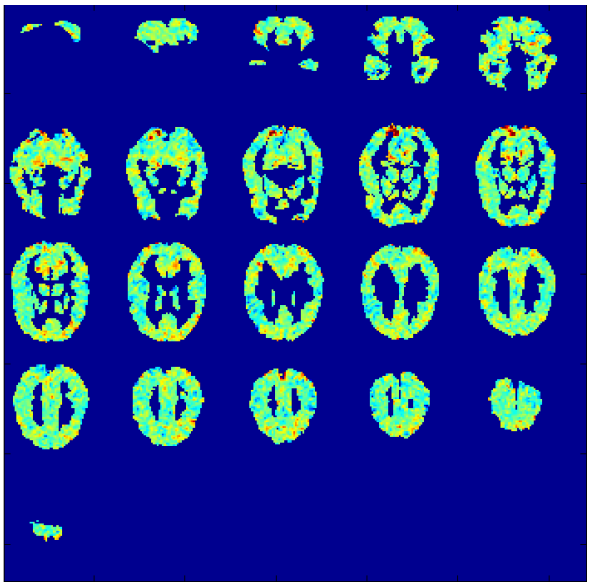


Figure 4.2.6: fMRI loadings for components 1 and 2 learned by the CCA-mult-WP method, for subjects 1WP (A) and 5WP. The brown ellipses on the component 1 figures highlight the right fusiform projections on one slice. As can be seen in the component 1 figures, strong projections are seen in other slices and also in the left fusiform.

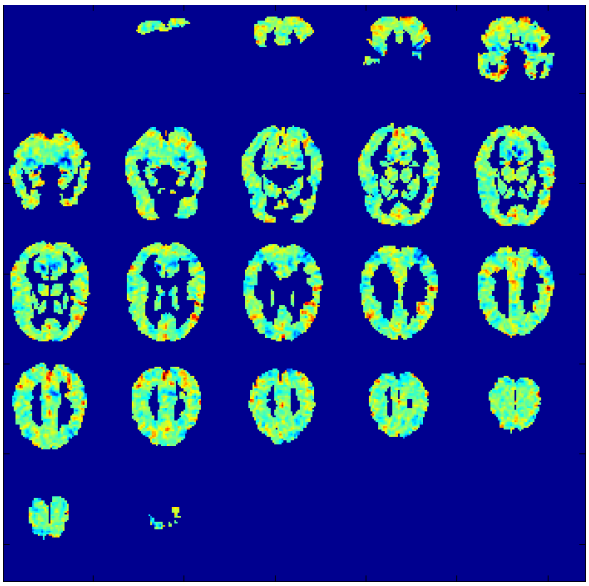
(a) subject 1WO, component 1



(b) subject 11WO, component 1



(c) subject 1WO, component 2



(d) subject 11WO, component 2

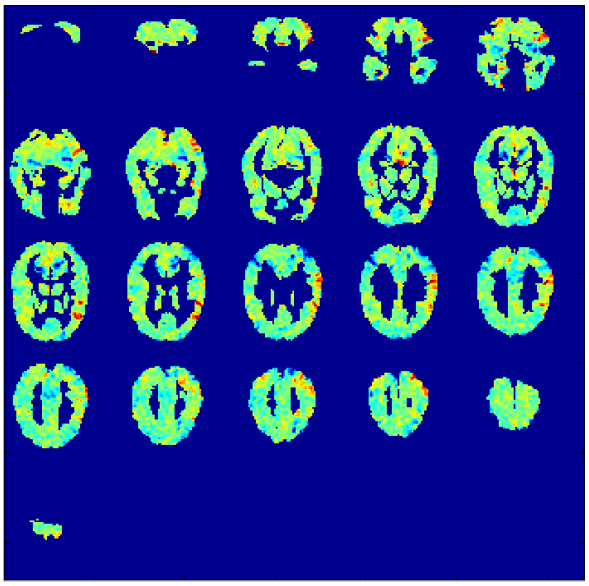
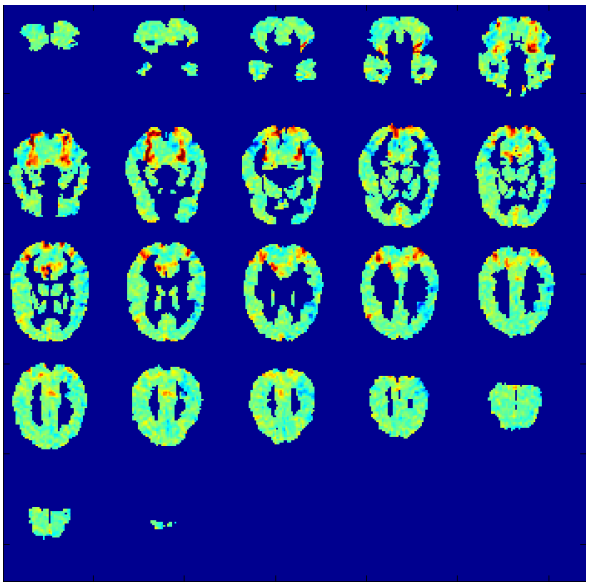
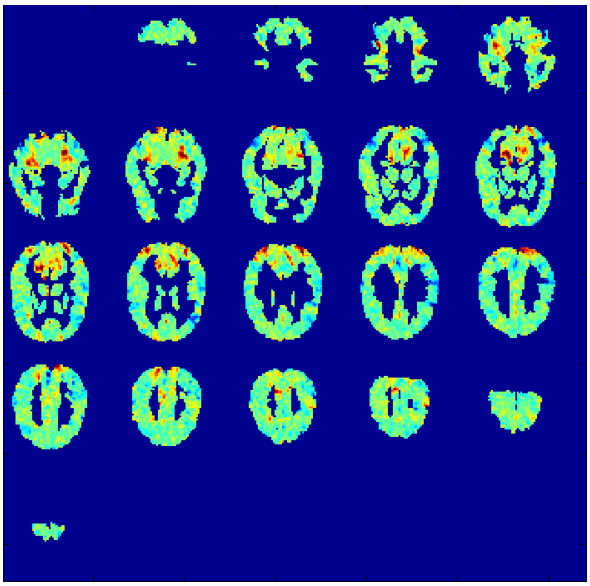


Figure 4.2.7: fMRI loadings for components 1 and 2 learned by the CCA-mult-WO method, for subjects 1WO (A) and 11WO.

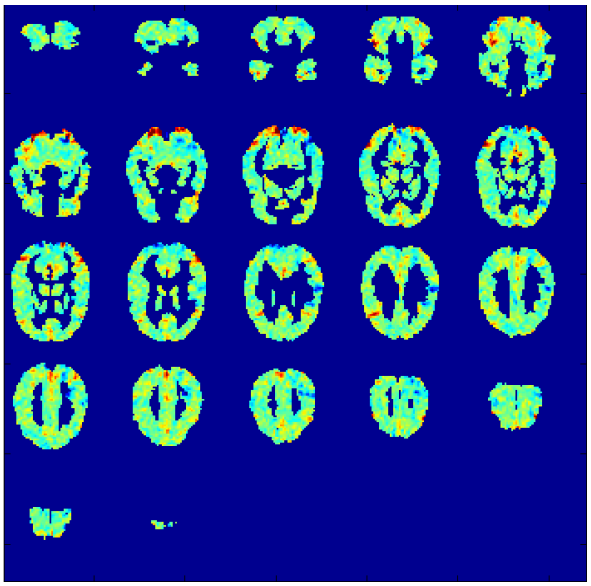
(a) subject 1WP, component 1



(b) subject 5WP, component 1



(c) subject 1WP, component 2



(d) subject 5WP, component 2

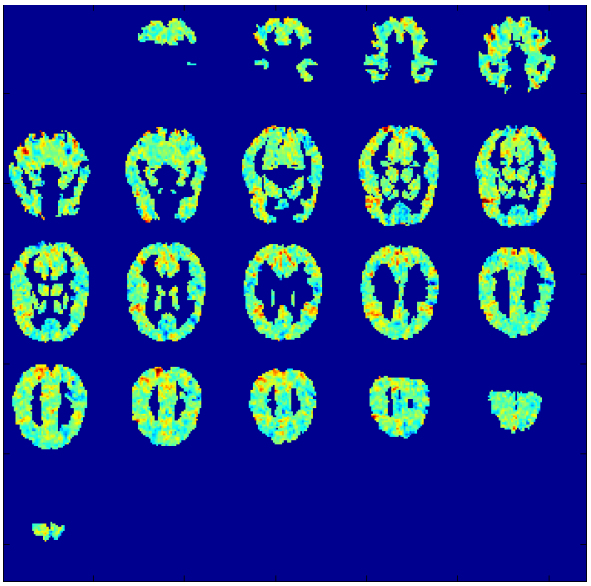
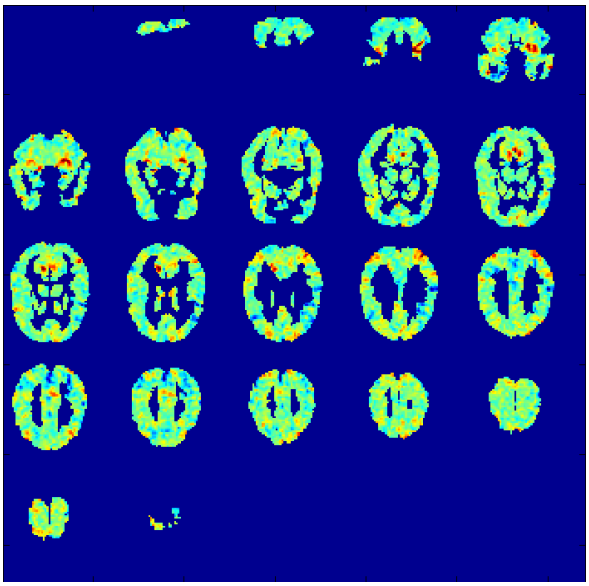
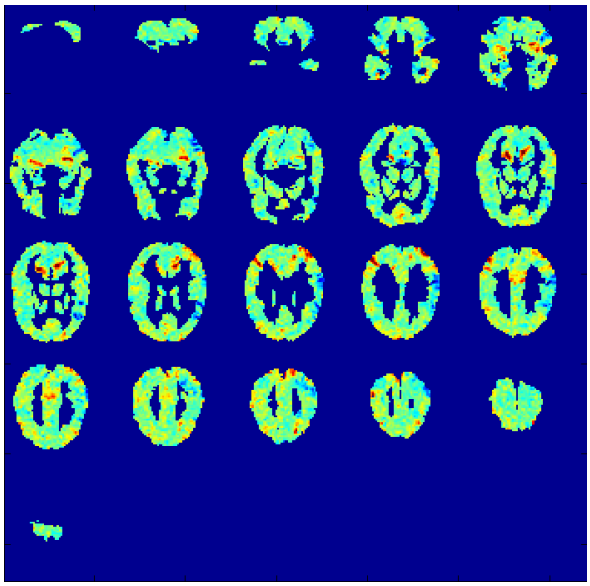


Figure 4.2.8: fMRI loadings for components 1 and 2 learned by the CCA-mult-comb method, for subjects 1WP (A) and 5WP.

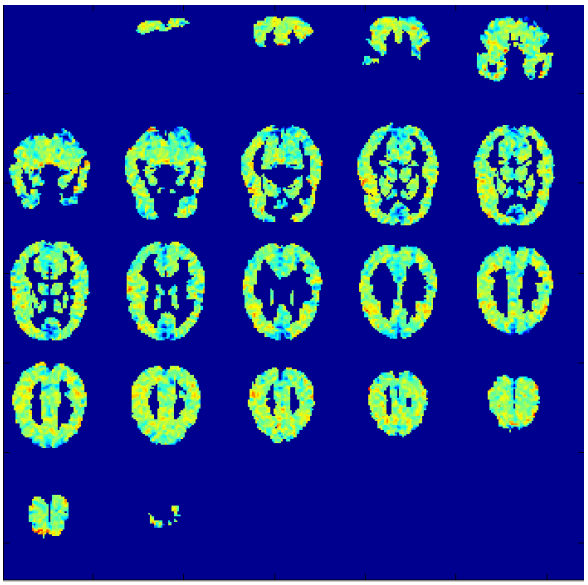
(a) subject 1WO, component 1



(b) subject 11WO, component 1



(c) subject 1WO, component 2



(d) subject 11WO, component 2

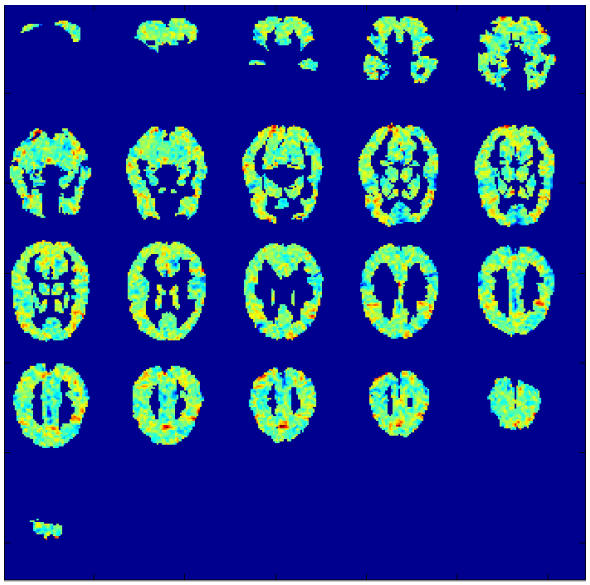


Figure 4.2.9: fMRI loadings for components 1 and 2 learned by the CCA-mult-comb method, for subjects 1WO (A) and 11WO.

Now we turn to the learned features when PCA is used. We first look at the stimulus-word rankings, shown in table 4.2.3. Comparing table 4.2.3 with table 4.2.1, we see that the rankings for the first learned feature are highly similar: we also see the shelter-tool dimension for the PCA-concat-WP and PCA-concat-comb methods and the word-length dimension for the PCA-concat-WO method. In some of the other cases, we also see similarities between the rankings shown in table 4.2.3 and the corresponding rankings shown in table 4.2.1.

	component 1	component 2	component 3	component 4	component 5
positive	knife	leg	cat	tomato	igloo
	screwdriver	key	beetle	cup	house
	spoon	glass	horse	house	car
	carrot	pants	butterfly	butterfly	truck
	celery	bottle	telephone	lettuce	ant
	saw	chisel	bee	apartment	carrot
	bottle	chair	fly	door	eye
	hammer	cat	dog	refrigerator	fly
	key	closet	eye	igloo	key
	pliers	dress	hand	celery	train
negative	apartment	dog	knife	arm	window
	barn	bear	screwdriver	foot	refrigerator
	closet	cow	church	coat	coat
	church	pliers	saw	hand	chisel
	house	spoon	hammer	leg	horse
	train	screwdriver	window	bicycle	telephone
	window	bed	spoon	closet	dog
	arch	ant	closet	table	pliers
	desk	carrot	chimney	desk	skirt
	bed	beetle	desk	telephone	dress

(a) PCA-concat-WP

	component 1	component 2	component 3	component 4	component 5
positive	apartment	screwdriver	cat	tomato	glass
	refrigerator	pliers	cow	door	key
	airplane	celery	fly	pliers	chimney
	dresser	carrot	dog	arch	train
	butterfly	lettuce	tomato	hammer	cup
	telephone	butterfly	cup	key	fly
	closet	knife	coat	house	leg
	house	telephone	bear	knife	celery
	bicycle	bottle	bell	chisel	church
	church	hammer	beetle	spoon	shirt
negative	eye	car	foot	shirt	bed
	leg	house	arm	lettuce	butterfly
	foot	desk	dresser	foot	eye
	arm	bed	hand	glass	bicycle
	saw	door	chair	bee	telephone
	bee	closet	desk	train	coat
	ant	barn	bicycle	dog	dresser
	cat	truck	leg	beetle	car
	key	church	screwdriver	carrot	pants
	cup	arch	closet	ant	knife

(b) PCA-concat-WO

	component 1	component 2	component 3	component 4	component 5
positive	apartment	leg	dog	cat	house
	house	eye	beetle	chisel	tomato
	closet	cat	cow	tomato	cup
	church	key	bear	butterfly	door
	barn	car	telephone	horse	igloo
	train	ant	butterfly	skirt	key
	desk	fly	lettuce	glass	carrot
	dresser	arm	fly	dress	car
	bed	bee	bee	telephone	cat
	arch	foot	ant	refrigerator	bell
negative	knife	screwdriver	leg	arm	foot
	screwdriver	refrigerator	closet	bed	hand
	carrot	pliers	chair	screwdriver	arm
	spoon	celery	bottle	spoon	telephone
	saw	apartment	door	car	bicycle
	celery	butterfly	glass	saw	coat
	pliers	knife	key	desk	dresser
	key	hammer	window	knife	shirt
	hammer	tomato	chisel	house	horse
	bottle	telephone	pants	ant	chair

(c) PCA-concat-comb

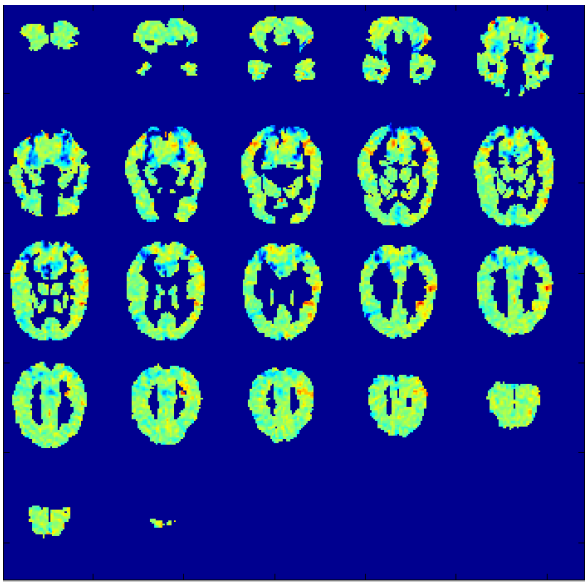
Table 4.2.3: The rankings of the the stimulus words in the first five components for the PCA-concat-WP (top left), PCA-concat-WO (top right), and PCA-concat-comb (bottom) methods.

We show the loadings for the first two learned features in figures 4.2.10, 4.2.11, 4.2.12, and 4.2.13, roughly equivalent to the loading figures in the CCA-mult cases. When the semantic dimensions present are similar between the PCA-concat cases and the CCA-mult cases (for instance the first component found by the two methods for the WP case), we see that the loadings are also similar.

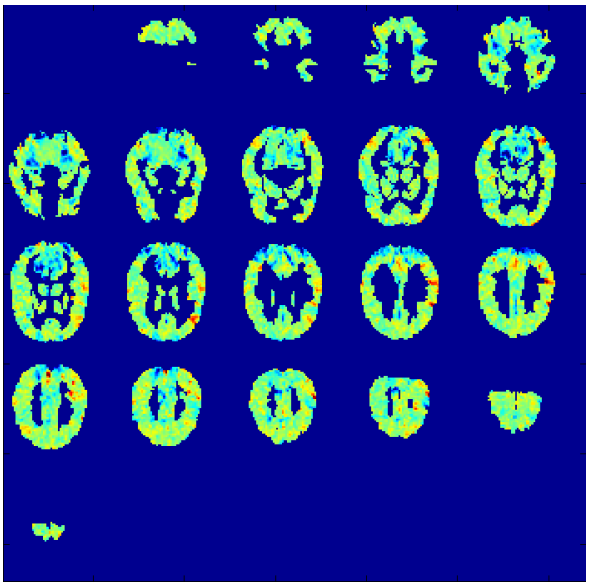
If there are similarities of the features learned in the PCA-concat cases compared to those learned in the CCA-mult cases, one might ask why the accuracies of these two groups of cases are quite different. We note that we present the information content of the learned features in relative terms without any indication about the actual values of the learned features. The distribution of the values of the learned features might give some indications about why the accuracies differ. In particular, we note that in the CCA-mult cases, the feature values are generally of lesser magnitude (for instance, mean=0, sd=0.13 for the first learned feature for the CCA-mult-comb method) compared to the feature values in the PCA-concat cases (for instance, mean=0, sd=18.44 for the first learned feature for the PCA-concat-comb method). This difference might affect the quality of the mapping from the base features, which in turn might influence the accuracies. The difference also highlights the effect of regularization in CCA, in which we penalize the norm of the loadings, which in turn affects the magnitude of the feature values; the poor results that we obtain when we use PCA

to learn the features might suggest that there is some over-fitting involved in the features learned using PCA, but not present when we use CCA due to the regularization.

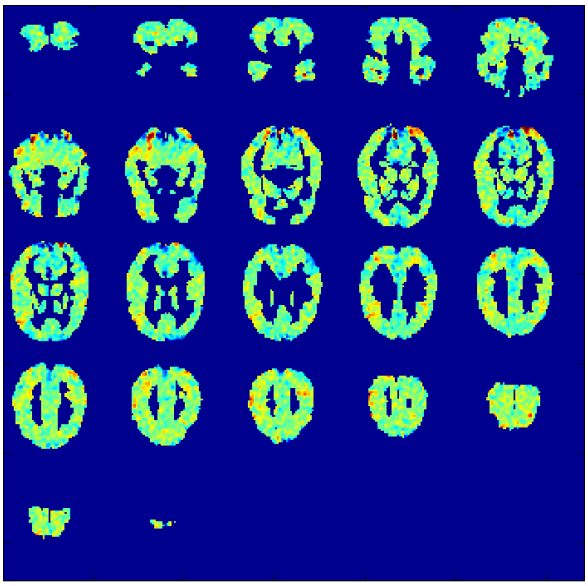
(a) subject 1WP, component 1



(b) subject 5WP, component 1



(c) subject 1WP, component 2



(d) subject 5WP, component 2

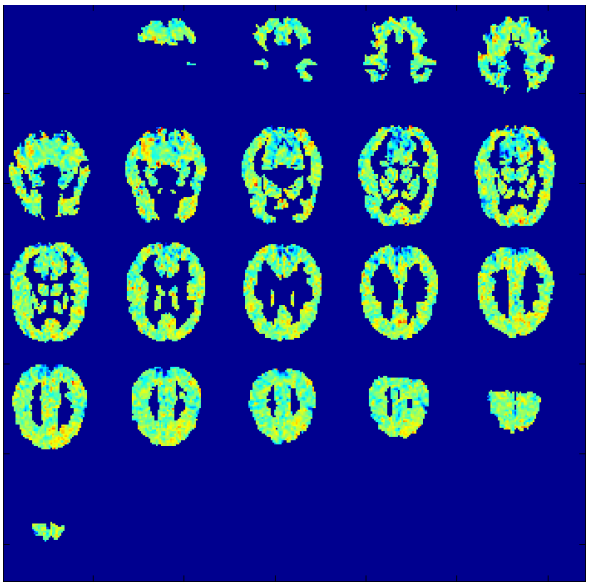
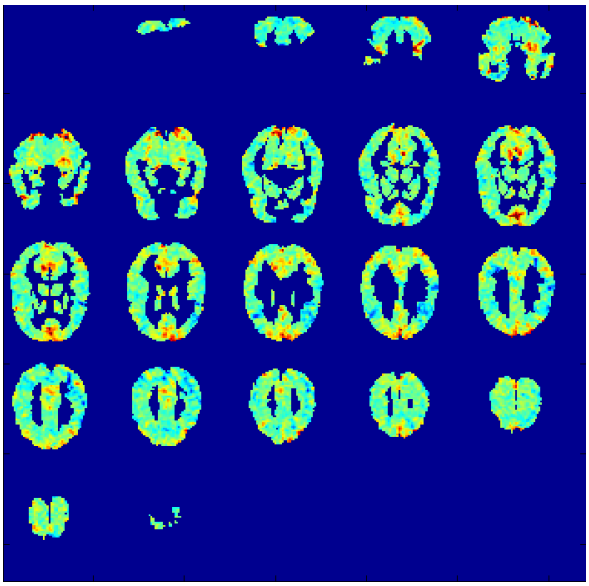
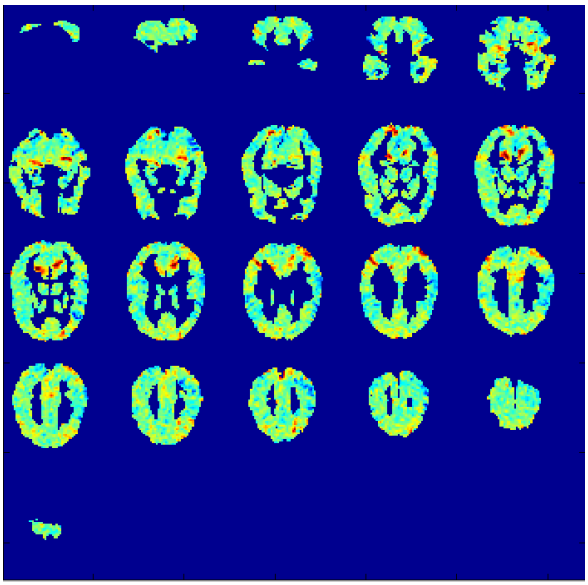


Figure 4.2.10: fMRI loadings for components 1 and 2 learned by the PCA-concat-WP method, for subjects 1WP (A) and 5WP.

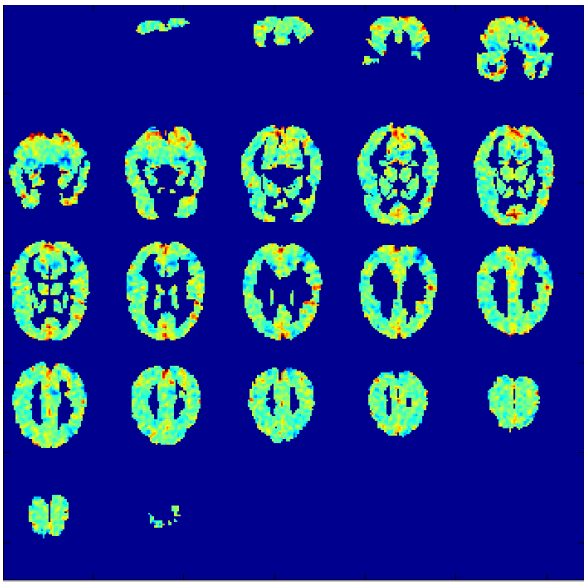
(a) subject 1WO, component 1



(b) subject 11WO, component 1



(c) subject 1WO, component 2



(d) subject 11WO, component 2

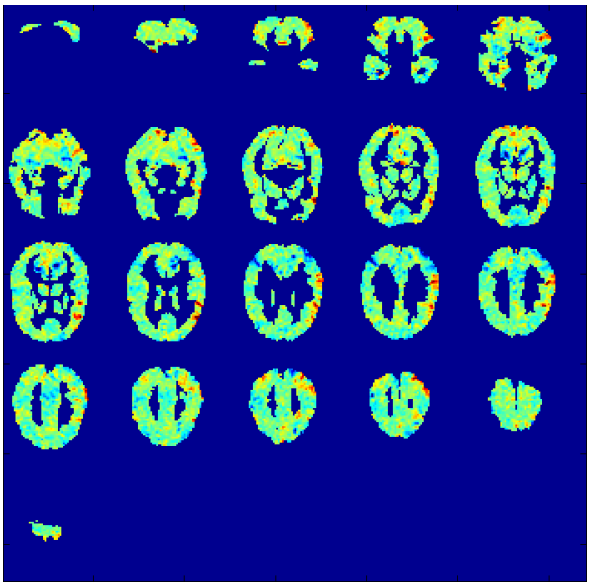
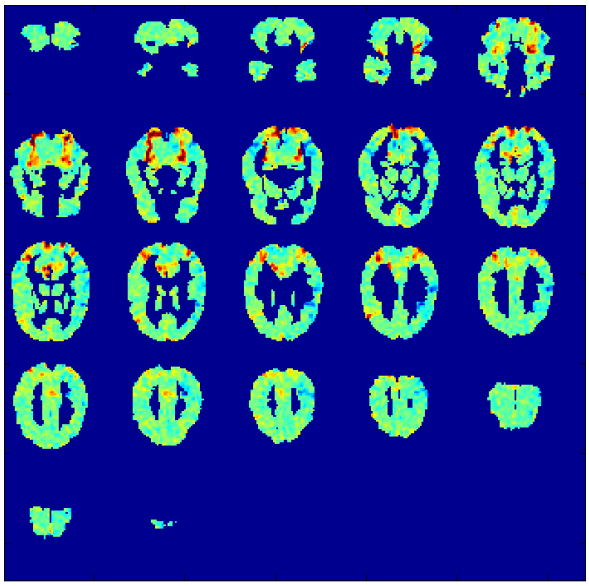
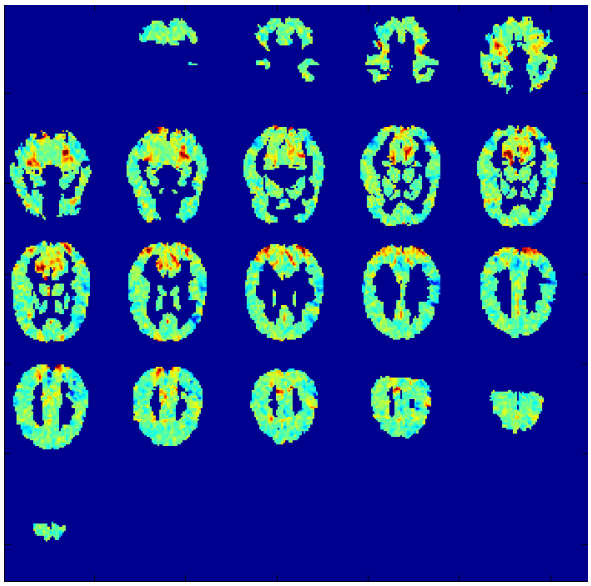


Figure 4.2.11: fMRI loadings for components 1 and 2 learned by the PCA-concat-WO method, for subjects 1WO (A) and 11WO.

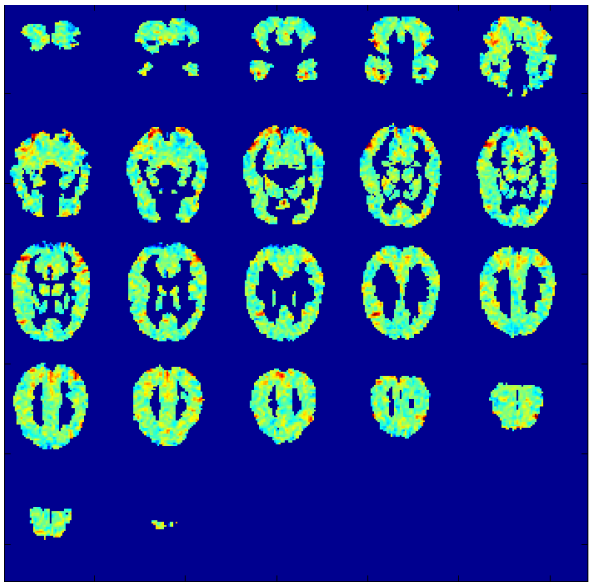
(a) subject 1WP, component 1



(b) subject 5WP, component 1



(c) subject 1WP, component 2



(d) subject 5WP, component 2

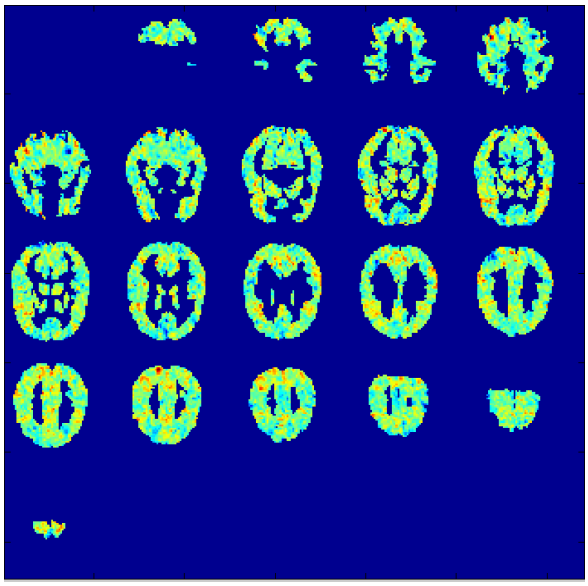
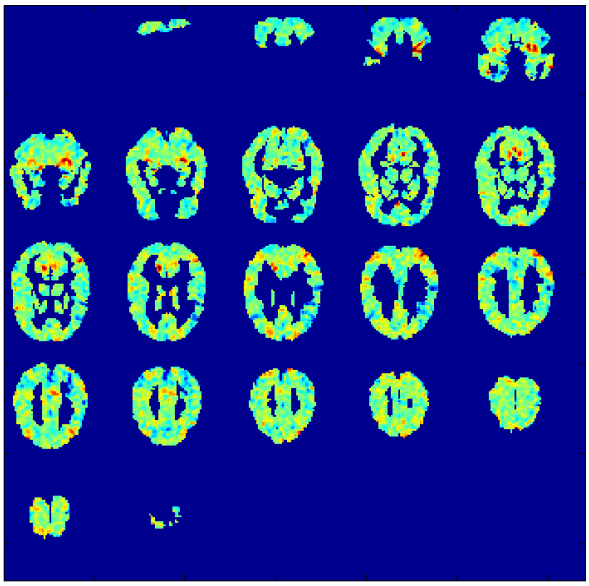
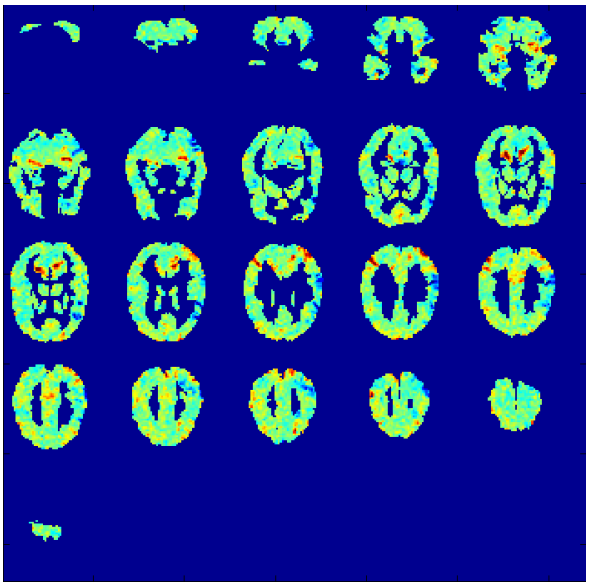


Figure 4.2.12: fMRI loadings for components 1 and 2 learned by the PCA-concat-comb method, for subjects 1WP (A) and 5WP.

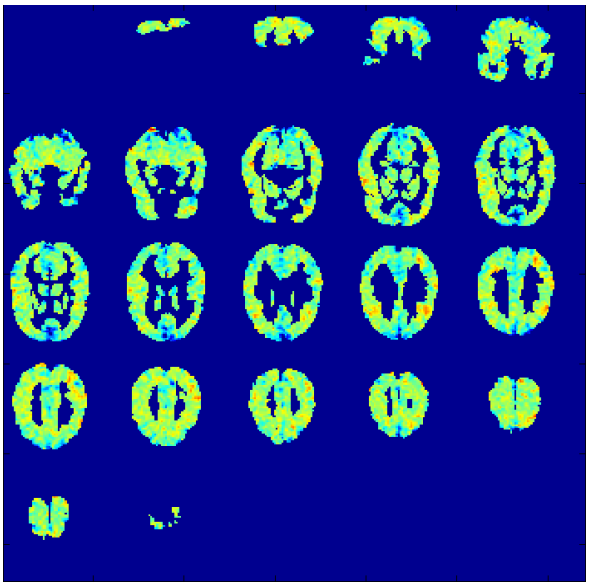
(a) subject 1WO, component 1



(b) subject 11WO, component 1



(c) subject 1WO, component 2



(d) subject 11WO, component 2

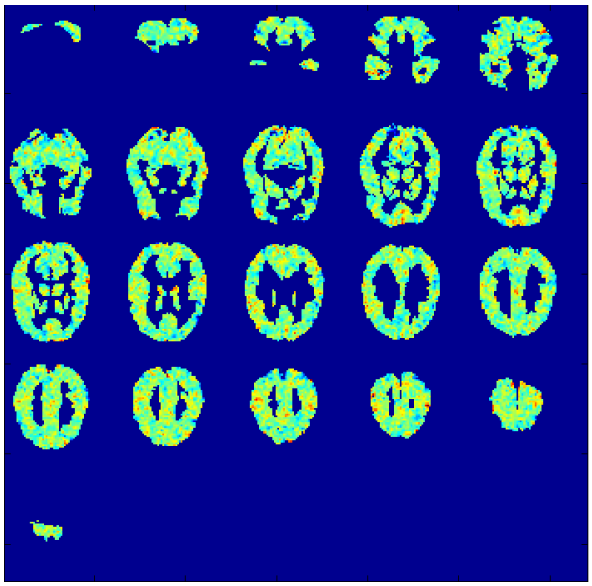


Figure 4.2.13: fMRI loadings for components 1 and 2 learned by the PCA-concat-comb method, for subjects 1WO (subject A) and 11WO.

4.2.6 Discussion

In light of the presented results, we consider the questions presented early and the answers suggested by the results.

1. Do we obtain better accuracies in the fMRI-common-feature model compared to the accuracies of the baseline model?

   As the results show, with enough components, the answer is yes when we use CCA to learn the common features.

2. If the answer to the previous question is yes, is the improvement in accuracies due to the integration of fMRI data across subjects and/or studies?

   For the PCA-indiv and CCA-indiv methods, we see that purely applying dimensionality reduction without any sharing of information across subjects and/or studies, we do not obtain improved accuracies relative to the baseline model's accuracies. This suggests that the answer to this question is yes, i.e. that the improvement in accuracies is due to the integration of fMRI data across subjects and/or studies, and not due solely to dimensionality reduction effects.

3. What is the number of learned common features that yields the best accuracies?

   Focusing on the CCA-mult variations, we see that the peak accuracies are reached when around 20 common features are used. Note that the optimal number of common features might be a function of the number of instances in the data (60 instances corresponding to the 60 words for the experiments described in this chapter) and the actual predictive task that we try to do.

4. What effect does integration across studies have, both in terms of the learned common features and in terms of accuracies?

   We do not see significant differences in peak accuracies when we integrate across studies compared with when we integrate within each study. However, when few common features are used, integration across studies often lead to better accuracies compared with integration only within study.

5. How is each learned common feature reflected on the brain for each subject?

   For a given feature, there are a lot of variations, depending on both the subject and the study. This also reflects the capability of the factor analysis approach in general in looking beyond the variations present in the datasets and finding the higher-order commonalities among the various datasets.

6. What kind of semantic information, if any, is present on the learned common features?

   A dimension consistently present in the learned common features is the separation between tool words and shelter words.

7. How many datasets are needed in order to obtain reliable learned common features, as measured by the resulting accuracies?

   Our results on applying the method to each of the three subjects that participated in both studies show that we can get improved accuracies even when we have only two datasets to integrate.

4.3 Experiments: some non-matching instances

In the second group of experiments, we investigate how effective the imputation method described in section 3.6 is when applied in the context of the fMRI-common-feature model when there are some non-matching instances. To do this, we leave out instances corresponding to a specific set of words (let us say, set A) from the data coming from the WP study, and we leave out instances corresponding to another specific set of words (let us say, set B) from the data coming from the WO study, and then we perform analysis on the resulting data. The two sets of words A and B form a pair, and we consider four different word-set pairs: unif-long (table 4.3.1), unif-short (table 4.3.2), cat-long (table 4.3.3), cat-short (table 4.3.4). *unif* indicates that the words in the two sets are chosen uniformly from each category while *cat* indicates that in each set, all words belonging to specific categories are left out; on the other hand, *long* indicates that the list of words in each set is relatively long compared to *short*. For the unif pairs, the words in each pair are chosen randomly, while for the cat pairs, the categories are chosen randomly.

Table 4.3.1: Words left out in the unif-long sets.

set 1	bear horse arm foot apartment house chimney window shirt skirt chair dresser butterfly fly glass spoon bell refrigerator chisel pliers carrot corn car truck
set 2	cat cow eye hand barn church arch door dress pants desk table ant bee cup knife telephone watch hammer saw celery lettuce airplane bicycle

Table 4.3.2: Words left out in the unif-short sets.

set 1	bear foot apartment chimney skirt dresser fly glass bell pliers corn truck
set 2	cat eye barn door pants table ant knife watch saw celery bicycle

Table 4.3.3: Words left out in the cat-long sets.

set 1	arm eye foot hand leg arch chimney closet door window coat dress pants shirt skirt ant bee beetle butterfly fly bell key refrigerator telephone watch chisel hammer pliers saw screwdriver
set 2	arm eye foot hand leg apartment barn church house igloo bed chair desk dresser table ant bee beetle butterfly fly bell key refrigerator telephone watch carrot celery corn lettuce tomato

We then run experiments based on each of these four pairs or groups. During the training process, for the data coming from the WP study, we leave out words based on one of the sets (set 1 or 2) and for the data coming from the WO study, we leave out words based on the set complementary to the one chosen for the WP data. In particular, we have the following two scenarios:

- WP-1, WO-1: leave out words from set 1 for the WP dataset and words from set 2 for the WO dataset.
- WP-2, WO-2: leave out words from set 2 for the WP dataset and words from set 1 for the WO dataset.

We also make references to *pair*-1 or *pair*-2 for a specific *pair* out of the four pairs, for instance, cat-short-1 and cat-short-2 for the cat-short pair. *pair*-1 (*pair*-2) refers to the case where we leave words from the pair's set 1 (set 2) for the data from the WP study and words from the pair's set 2 (set 1) for the data from the WO study.s

We seek to answer the following questions:

1. How does leaving the words affect the accuracies of the baseline model?

Table 4.3.4: Words left out in the cat-short sets.

set 1	arch chimney closet door window ant bee beetle butterfly fly chisel hammer pliers saw screwdriver
set 2	apartment barn church house igloo bed chair desk dresser table ant bee beetle butterfly fly

2. Does the imputation scheme yield better accuracies compared to methods relying on only the shared instances?
3. Does the imputation scheme yield better accuracies compared to the baseline model and the within-study common-factor approaches?
4. How do the accuracies vary with the number of nearest neighbors used?
5. How do the imputed values compare to the actual values?
6. What is the distribution of the sources for the imputed values (in terms of subjects and locations)?

4.3.1 fMRI Datasets

The WP and WO datasets described in sections 1.3.3 and 1.3.4 are used. We refer to these sections for descriptions about how the data were pre-processed.

4.3.2 Predefined semantic features

We use the 485verb and intel218 features, described in section 4.2.2.

4.3.3 Evaluation

We use the evaluation procedure described in section 4.2.3.

4.3.4 Methods

In all the experiments described here, we consider data from the WP and WO studies jointly. For this group of experiments, the following methods are tried:

- **LR**, the baseline model of Mitchell et al. (2008), where for each dataset we consider only the instances available to that dataset
- **PCA-shared**, an instantiation of the fMRI-common-feature model where we consider only the instances shared by all the datasets from both studies (WP and WO), concatenating the data matrices and applying PCA on the concatenated matrix to learn the common features across the two studies WP and WO
- **PCA-concat**, an instantiation of the fMRI-common-feature model where, for each study (WP or WO), we concatenate the data matrices coming from that study, using all the instances present for the study, and applying PCA on the concatenated matrix to obtain study-specific common features
- **PCA-*k*nn**, an instantiation of the fMRI-common-feature model where for each study (WP or WO), we impute the data for the instances missing for that study using k nearest neighbors, and we concatenate the resulting data matrices from both studies and applying PCA to the concatenated matrix to obtain the common features across the two studies; we consider $k = 5, 10, 20$, so we have the variations **PCA-5nn**, **PCA-10nn**, and **PCA-20nn**

- **CCA-shared**, an instantiation of the fMRI-common-feature model where we consider only the instance shared by all the datasets from both studies (WP and WO), applying CCA to the data matrices to obtain common features across the two studies
- **CCA-mult**, an instantiation of the fMRI-common-feature model where for each study (WP or WO), we apply CCA to the data matrices for that study, using all the instances present for that study, to obtain study-specific common features
- **CCA-*k*nn**, an instantiation of the fMRI-common-feature model where for each study (WP or WO), we impute the data for the instances missing for that study using k nearest neighbors, and we apply CCA to the resulting data matrices for both studies to learn common features across the two studies; as for PCA-knn, we consider $k = 5, 10, 20$, so we have the variations **CCA-5nn**, **CCA-10nn**, and **CCA-20nn**

All the above methods are run with all the subjects from both WP and WO studies, with a set of words left out for the data from the WP study and another set of words left out for the data from the WO study. In addition, we also perform experiments on the three shared subjects that we focus on in the previous group of experiments (subjects A, B, and C). For each of these three subjects, we apply our methods taking the data from the WP and WO studies only for that particular subject. In other words, in contrast with when we consider all the subjects from the two studies in which we have 20 datasets in total, in this latter case, the methods consider only two datasets, one dataset from each study for the subject of interest. This set of experiments give us insight into the effectiveness of the imputation scheme when there are only a few datasets to integrate.

For all the methods besides the baseline model LR, we use 10 learned common features.

4.3.5 Results

4.3.5.1 Accuracies

Imputing using data from multiple subjects We first look at the accuracies of the various methods when we consider all the 20 subjects from both WP and WO studies. These are shown in four figures:

- figure 4.3.1, mean accuracies for the subjects in each of the WP and WO studies when we use the unif groups of words with the 485verb features
- figure 4.3.2, mean accuracies for the subjects in each of the WP and WO studies when we use the cat groups of words with the 485verb features
- figure 4.3.3, mean accuracies for the subjects in each of the WP and WO studies when we use the unif groups of words with the intel218 features
- figure 4.3.4, mean accuracies for the subjects in each of the WP and WO studies when we use the cat groups of words with the intel218 features

Besides the methods listed in section 4.3.4, whose accuracies are shown as bars in the figures, we also have accuracies of additional methods represented as lines:

- **LR-full**, the baseline method of Mitchell et al. (2008), with no missing instances
- **PCA-full**, an instantiation of the fMRI-common-feature model, using PCA to learn the common features across the two datasets, with no missing instances
- **CCA-full**, an instantiation of the fMRI-common-feature model, using CCA to learn the common features across the two datasets, with no missing instances

We obtain the results for the additional methods from the first group of experiments in section 4.2, and they are shown to highlight the difference between when some of the words are left out versus when all the words are present.

Let us first consider how the accuracies of the imputation methods compare with those of the baseline method LR. We see that the accuracies of the CCA-knn methods are comparable or significantly better compared to the accuracies of the LR method. On the other hand, the accuracies of the PCA-knn methods in the various cases are at best comparable to the accuracies of the baseline method. This is not different

from the trend we see in the results of section 4.2, where CCA-based methods can outperform the baseline method while the PCA-based can at best yield comparable performance to the LR method.

Next, we look at how the imputation methods perform compared to the shared methods. Remember that in the shared methods, we learn common features based on only the shared instances, so these methods cannot use instances that are not present in both studies. The figures show that the accuracies of the CCA-knn methods are also comparable or significantly better compared to the accuracies of their counterpart method CCA-shared. This is also the case with the PCA-knn methods and their counterpart method PCA-shared. This means that doing the imputation is indeed better compared to relying on only the shared instances.

So far we are considering all 20 subjects from both WP and WO studies. In section 4.2, we see that we can improve on the baseline LR method's accuracies even when we combine data from subjects only within a particular study, not combining data across studies; in particular, the improvements are seen especially when we use the CCA-mult method. Since all the subjects from a study share the same instances, we can also perform the within-study analysis on the subjects from a particular study, WP or WO, applying the CCA-mult method or the PCA-concat method. The resulting accuracies, labeled with CCA-mult and PCA-concat, are also shown in the figures. When we compare the accuracies of the CCA-knn methods with those of the CCA-mult method, in a majority of the cases, the accuracies are comparable. However, in a few cases, the accuracies of the CCA-knn methods are better, and in a few other cases, the accuracies of the CCA-knn methods are worse. The cases where the CCA-knn methods outperform the CCA-mult method are

- 485verb-unif-short (WO-2)
- 485verb-cat-short (WP-2)
- intel218-unif-short (WO-2)
- intel218-cat-short (WP-2)

while the cases where the CCA-knn methods underperform the CCA-mult method are

- 485verb-cat-long (WP-1, WP-2, WO-1, WO-2)
- 485verb-cat-short (WP-1, WO-1)
- intel218-unif-long (WO-2)
- intel218-cat-long (WO-1)

Note that these are out of a total of 32 cases. In other words, in 4 out of 32 cases, the CCA-knn methods outperform the CCA-mult method while in 8 out of 32 cases, the CCA-knn methods underperform the CCA-mult method.

We can also see the cases where the PCA-knn methods outperform the PCA-concat method:

- 485verb-unif-long (WP-1, WO-2)
- 485verb-unif-short (WP-1, WO-1, WO-2)
- 485verb-cat-short (WO-1, WP-2, WO-2)
- intel218-unif-long (WP-1, WO-1)
- intel218-unif-short (WO-1, WO-2)
- intel218-cat-short (WO-1, WP-2, WO-2)

and the cases where the PCA-knn methods underperform the PCA-concat method:

- 485verb-cat-long (WP-1)
- intel218-cat-long (WP-1)

To summarize, in 15 out of 32 cases, the PCA-knn methods outperform the PCA-concat method, while in 2 out of 32 cases, the PCA-knn methods underperform the PCA-concat method.

Focusing on the results for the methods that use CCA, since they give the better accuracies, we see that in some cases, the information obtained through the imputation procedure can improve the features learned compared to when the features are learned using only the within-study information, but in other

cases, the information actually harms the learned features. Nonetheless, for these CCA-based methods, in a majority of cases, we do not see significant differences between the accuracies of the CCA-knn methods and those of the CCA-mult method.

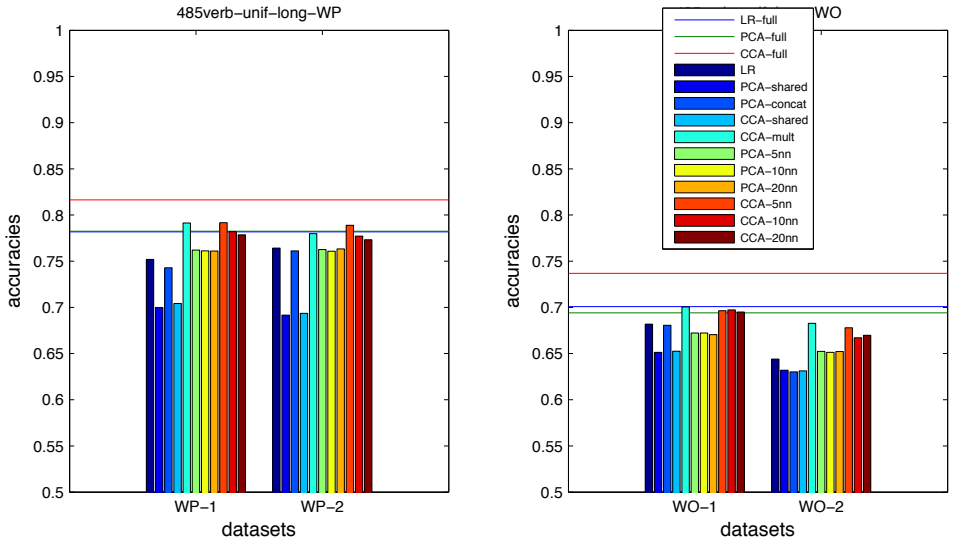
Now we consider how the accuracies of the imputation methods compared with those of the original methods having data for all the words. The expectation is that since the imputation methods have access to less information, their accuracies will be worse compared to those of the full methods. Nonetheless, we do see accuracies of the CCA-knn methods being better than those of the LR-full method, namely in the following 7 cases (again out of 32 total cases):

- 485verb-unif-short (WP-1, WO-2)
- intel218-unif-long (WP-1)
- intel218-unif-short (WP-1, WP-2)
- intel218-cat-short (WP-1, WP-2)

So it is definitely possible to obtain improvements in accuracies over those of the baseline method, using the imputation method in conjunction with CCA. In contrast, there are no cases where the methods that use imputation outperform the CCA-full method.

One might ask how the number of nearest neighbors affect the accuracies. In most cases, the variation across the number of nearest neighbors is minimal, although in some cases, we do see the accuracies marginally decrease as the number of nearest neighbors increases. The trend, however, does not appear to be significant or consistent.

(a) unif-long



(b) unif-short

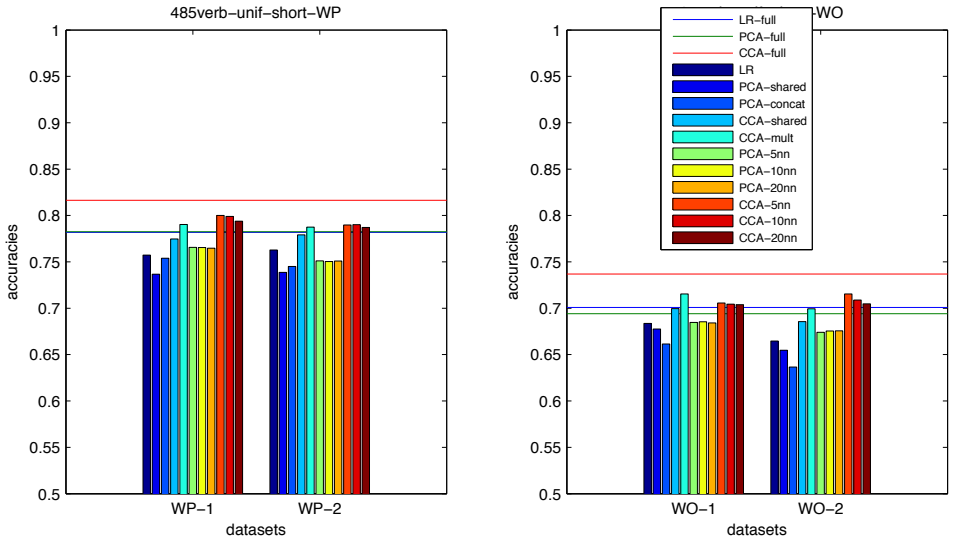
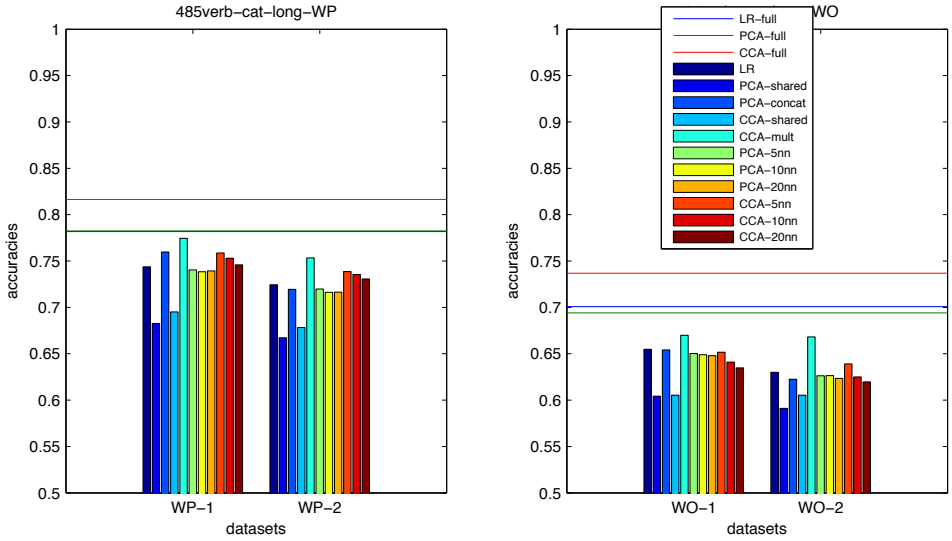


Figure 4.3.1: Mean accuracies when we use the 485verb features and leaving words from the unif-long (top row) and unif-short (bottom row) sets. All the 60 words are considered in LR-full, PCA-full, and CCA-full. Methods that utilize imputation for missing instances are PCA-5nn, PCA-10nn, PCA-20nn, CCA-5nn, CCA-10nn, and CCA-20nn.

(a) cat-long



(b) cat-short

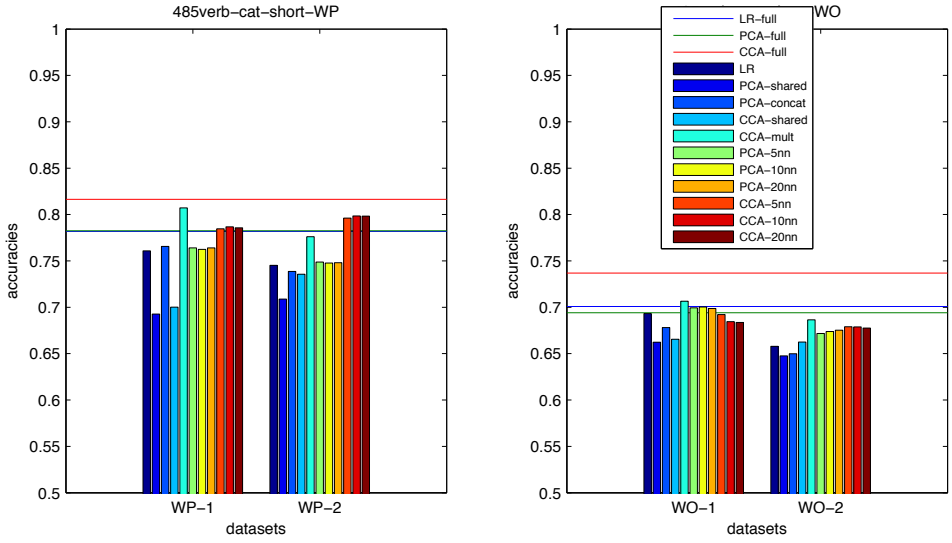
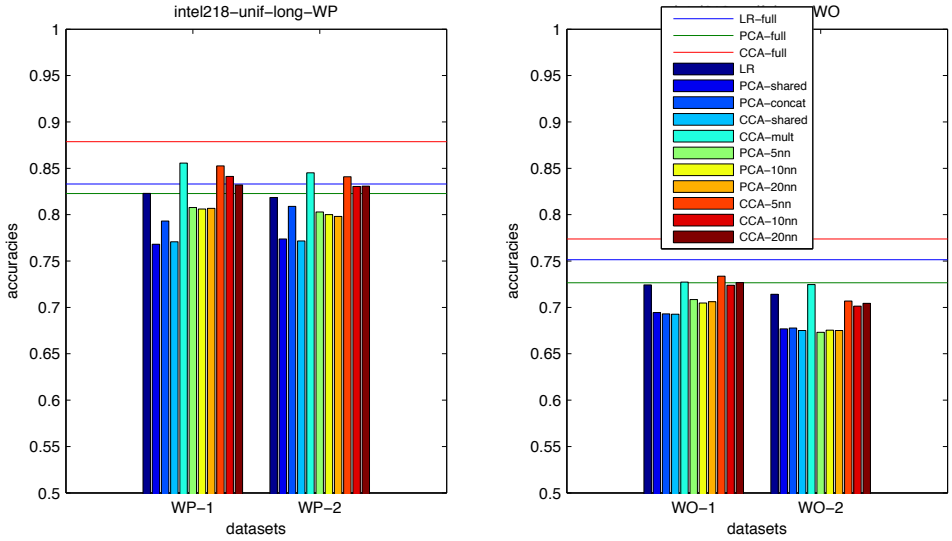


Figure 4.3.2: Mean accuracies when we use the 485verb features and leaving words from the cat-long (top row) and cat-short (bottom row) sets.

(a) unif-long



(b) unif-short

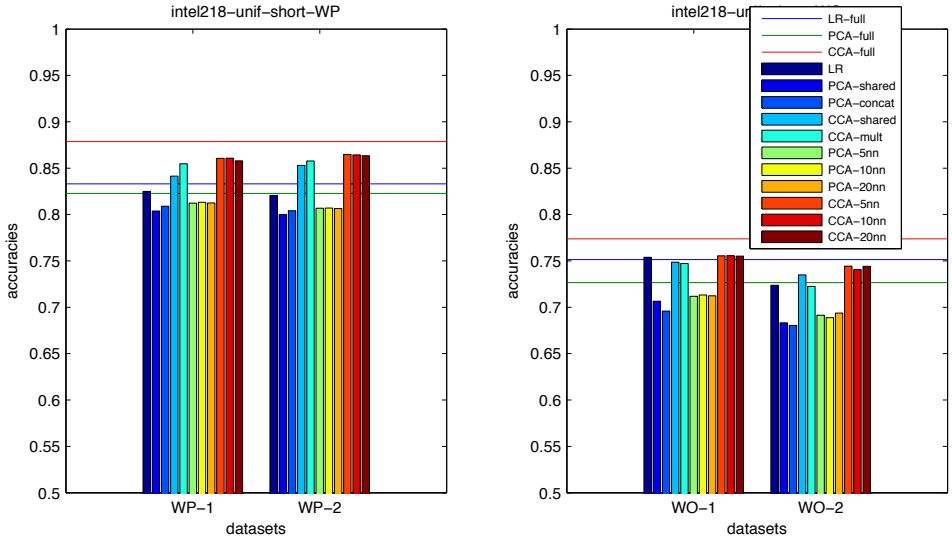
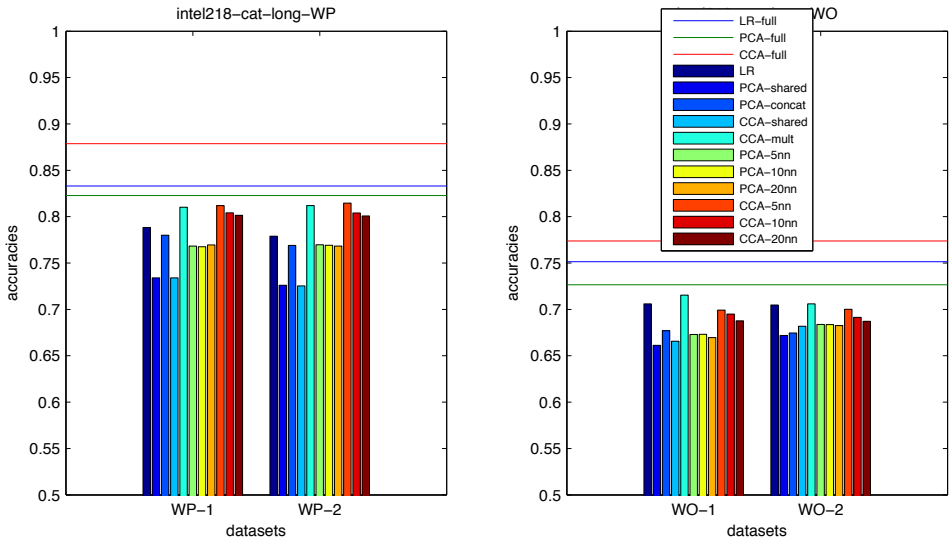


Figure 4.3.3: Mean accuracies when we use the intel218 features and leaving words from the unif-long (top row) and unif-short (bottom row) sets.

(a) cat-long



(b) cat-short

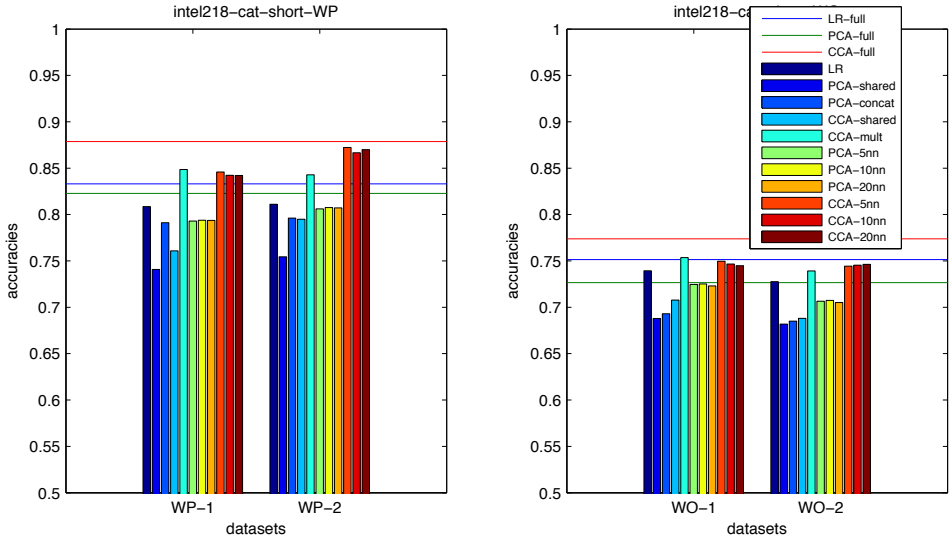


Figure 4.3.4: Mean accuracies when we use the intel218 features and leaving words from the cat-long (top row) and cat-short (bottom row) sets.

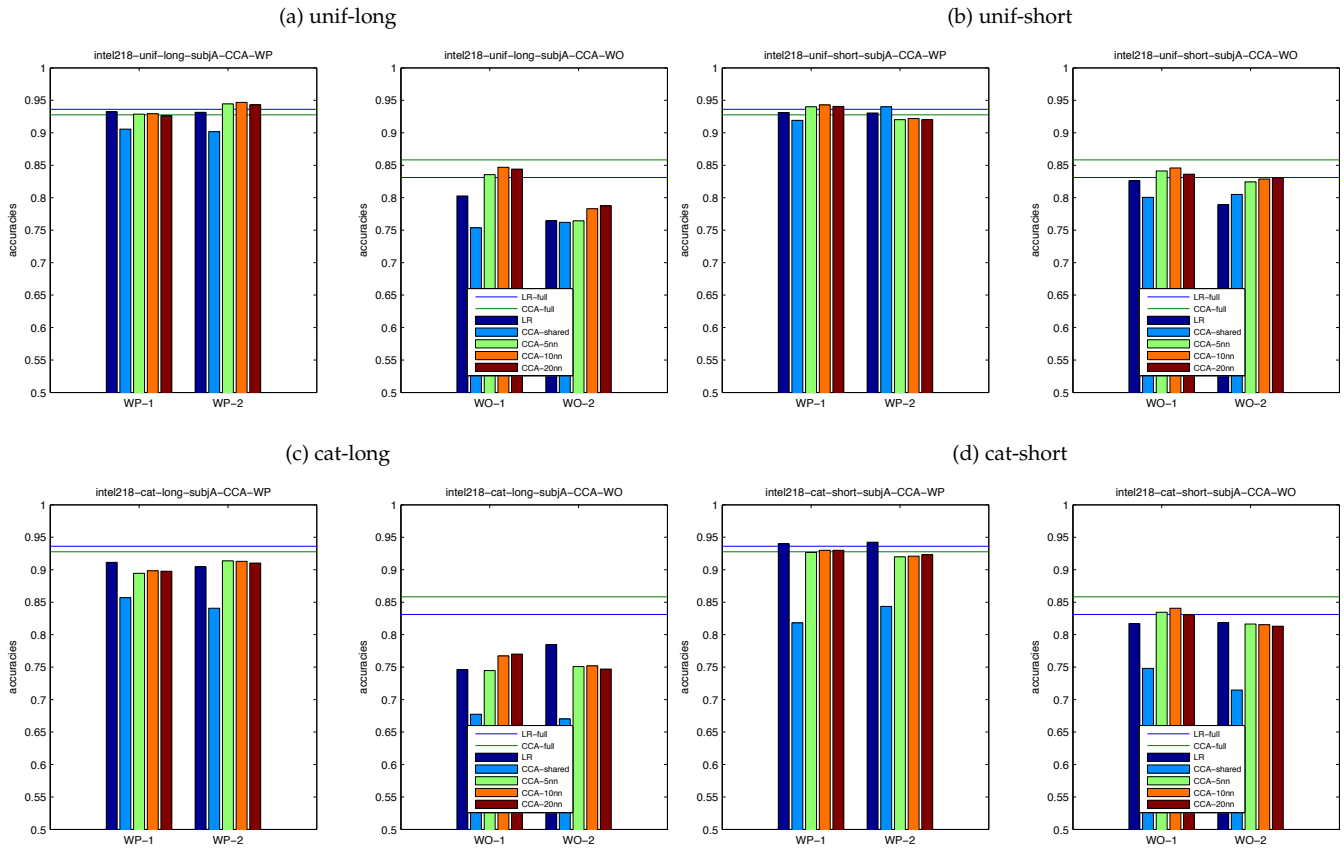


Figure 4.3.5: Accuracies for subject A when we use the intel218 features and leaving out words based on the available sets.

Imputing using data from only a single subject Next, we look at how the imputation methods perform when each of the two studies contains only a single subject. As was done in section 4.2, we consider those subjects who participated in both studies and we consider each of these three subjects separately. To avoid a clutter of figures, we show the results only for the intel218 features and only for the CCA-based methods. Note that because there is only one subject for each study, we cannot apply the CCA-mult method in these cases. The results are shown in three figures:

- figure 4.3.5 for subject A
- figure 4.3.6 for subject B
- figure 4.3.7 for subject C

We consider each subject separately. There are 16 cases in total for each subject. For subject A, when comparing the accuracies of the CCA-knn methods with those of the baseline LR method, we see in the following eight cases the CCA-knn methods outperforming the LR method:

- intel218-unif-long (WP-2, WO-1, WO-2)
- intel218-unif-short (WP-1, WO-1, WO-2)
- intel218-cat-long (WO-1)
- intel218-cat-short (WO-1)

and in the following five cases, the CCA-knn methods underperforming the LR method:

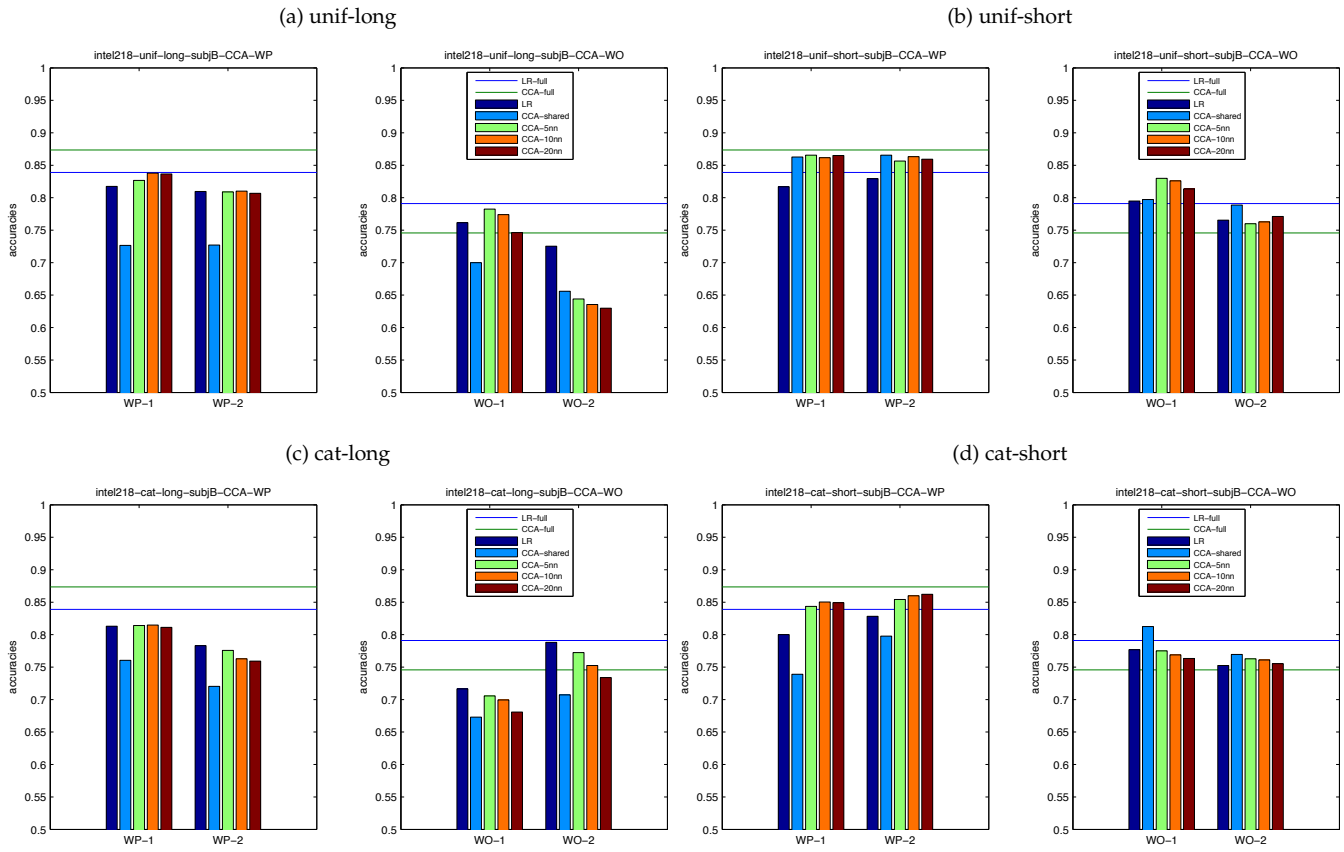


Figure 4.3.6: Accuracies for subject B when we use the intel218 features and leaving out words based on the available sets.

- intel218-unif-short (WP-2)
- intel218-cat-long (WP-1, WO-2)
- intel218-cat-short (WP-1, WP-2)

On the other hand, when comparing the accuracies of the CCA-knn methods with those of the CCA-shared method, we see in most cases the CCA-knn methods outperforming the CCA-shared method, with the exception of the case intel218-unif-short (WP-2).

For subject B, in the following eight cases the CCA-knn methods outperform the LR method:

- intel218-unif-long (WP1, WO-1)
- intel218-unif-short (WP-1, WP-2, WO-1)
- intel218-cat-short (WP-1, WP-2, WO-2)

and in these three cases, the accuracies of the CCA-knn methods are worse compared to those of the baseline LR method:

- intel218-unif-long (WO-2)
- intel218-cat-long (WO-1, WO-2)

When we compare the accuracies of the CCA-knn methods with those of the CCA-shared method for this particular subject, in a majority of cases, the accuracies of the CCA-knn methods are comparable or better compared to the accuracies of the CCA-shared method. The exceptions are

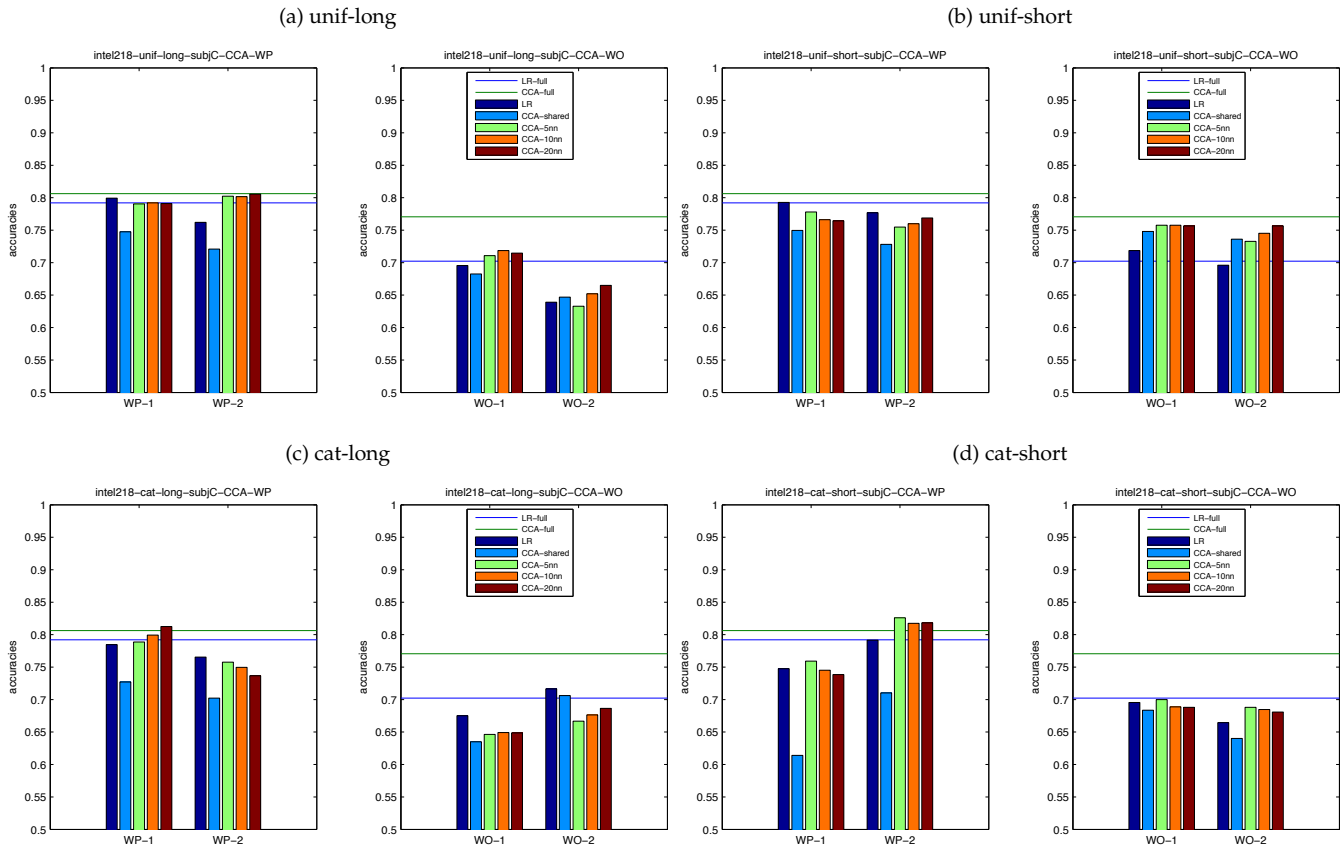


Figure 4.3.7: Accuracies for subject C when we use the intel218 features and leaving out words based on the available sets.

- intel218-unif-long (WO-2)
- intel218-unif-short (WO-2)
- intel218-cat-short (WO-1)

And for subject C, in the following nine cases the CCA-knn methods outperform the LR method:

- intel218-unif-long (WP-2, WO-1, WO-2)
- intel218-unif-short (WO-1, WO-2)
- intel218-cat-long (WP-1)
- intel218-cat-short (WP-1, WP-2, WO-2)

and in these three cases, the accuracies of the CCA-knn methods are worse compared to those of the baseline LR method:

- intel218-unif-short (WP-1)
- intel218-cat-long (WO-1, WO-2)

When we compare the accuracies of the CCA-knn methods with those of the CCA-shared method for this particular subject, in a majority of cases, the accuracies of the CCA-knn methods are comparable or better compared to the accuracies of the CCA-shared method, with the exception of intel218-cat-long (WO-2).

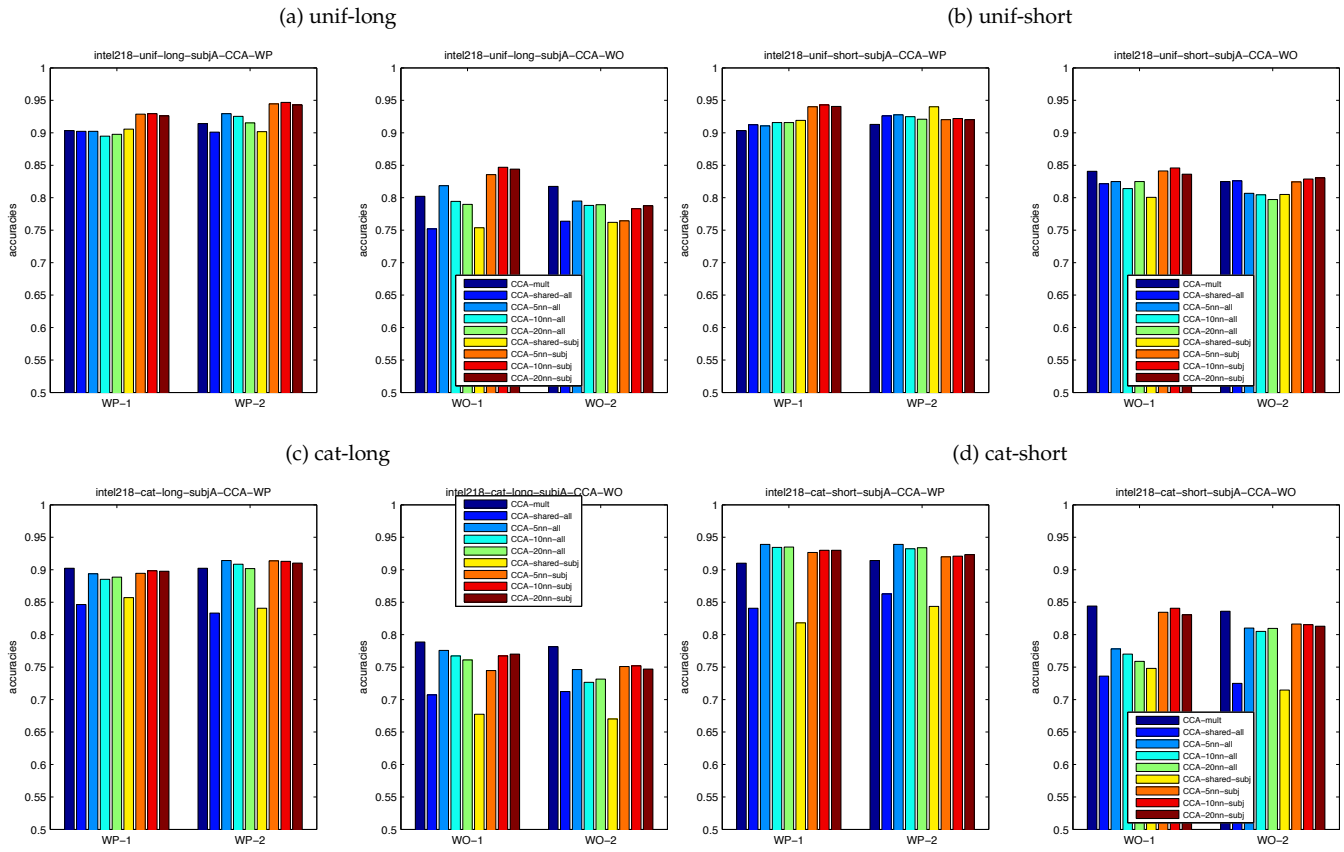


Figure 4.3.8: Accuracies for subject A when using all the subjects in both studies vs when we have only subject A in each study, using the intel218 features and leaving out words based on the available sets.

To summarize, for all three subjects, there are more cases (25 out of 48 cases) where the CCA-knn method leads to better accuracies compared to those of the LR method than cases (11 out of 48 cases) where the CCA-knn method leads to worse accuracies. This is also the case when we compare the CCA-knn method with the CCA-shared methods.

Comparing multiple-subject imputation vs single-subject imputation One might wonder how the accuracies for each of these three subjects differ when we use the data from the same subject from the other study versus when also have data for other subjects in the other study. To answer this question, for each of subjects A, B, and C, we plot the accuracies of the impute variations in these two cases in the following figures:

- figure 4.3.8 for subject A
- figure 4.3.9 for subject B
- figure 4.3.10 for subject C

In these figures, we also show the accuracies when we use the CCA-mult method, that is when we use data from the other subjects in the same study but not the subjects from the other study. We consider the accuracies in two groups:

- subj accuracies: the accuracies of methods whose labels end with nn-subj, indicating the use of data from the same subject in the other study to perform the imputation

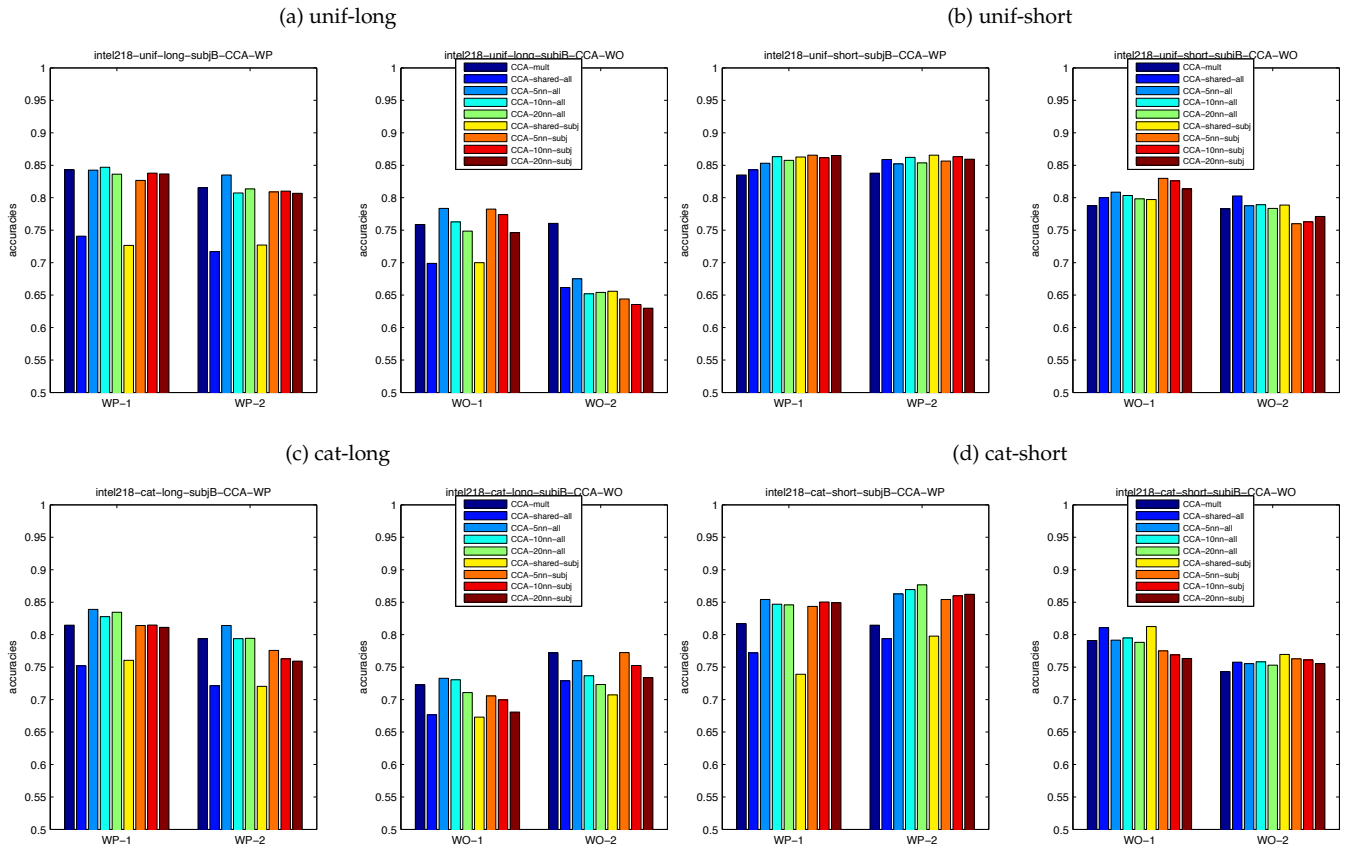


Figure 4.3.9: Accuracies for subject B when using all the subjects in both studies vs when we have only subject B in each study, using the intel218 features and leaving out words based on the available sets.

- all accuracies: the accuracies of methods whose labels end with nn-all, indicating the use of data from all the subjects in the other study to perform the imputation

Let us first consider the results for subject A. In the following cases (4 out of 16), the subj accuracies outperform the all accuracies:

- intel218-unif-long (WP-1, WP-2, WO-1)
- intel218-unif-short (WP-1)

and in these cases (3 out of 16), the subj accuracies underperform the all accuracies.

- intel218-unif-long (WO-2)
- intel218-cat-long (WO-1)
- intel218-cat-short (WP-2)

For subject B, in one case, the subj accuracies outperform the all accuracies:

- intel218-unif-short (WO-1)

while in 8 out of 16 cases, the subj accuracies underperform the all accuracies.

- intel218-unif-long (WP-2, WO-2)
- intel218-unif-short (WO-2)
- intel218-cat-long (WP-1, WP-2, WO-1)

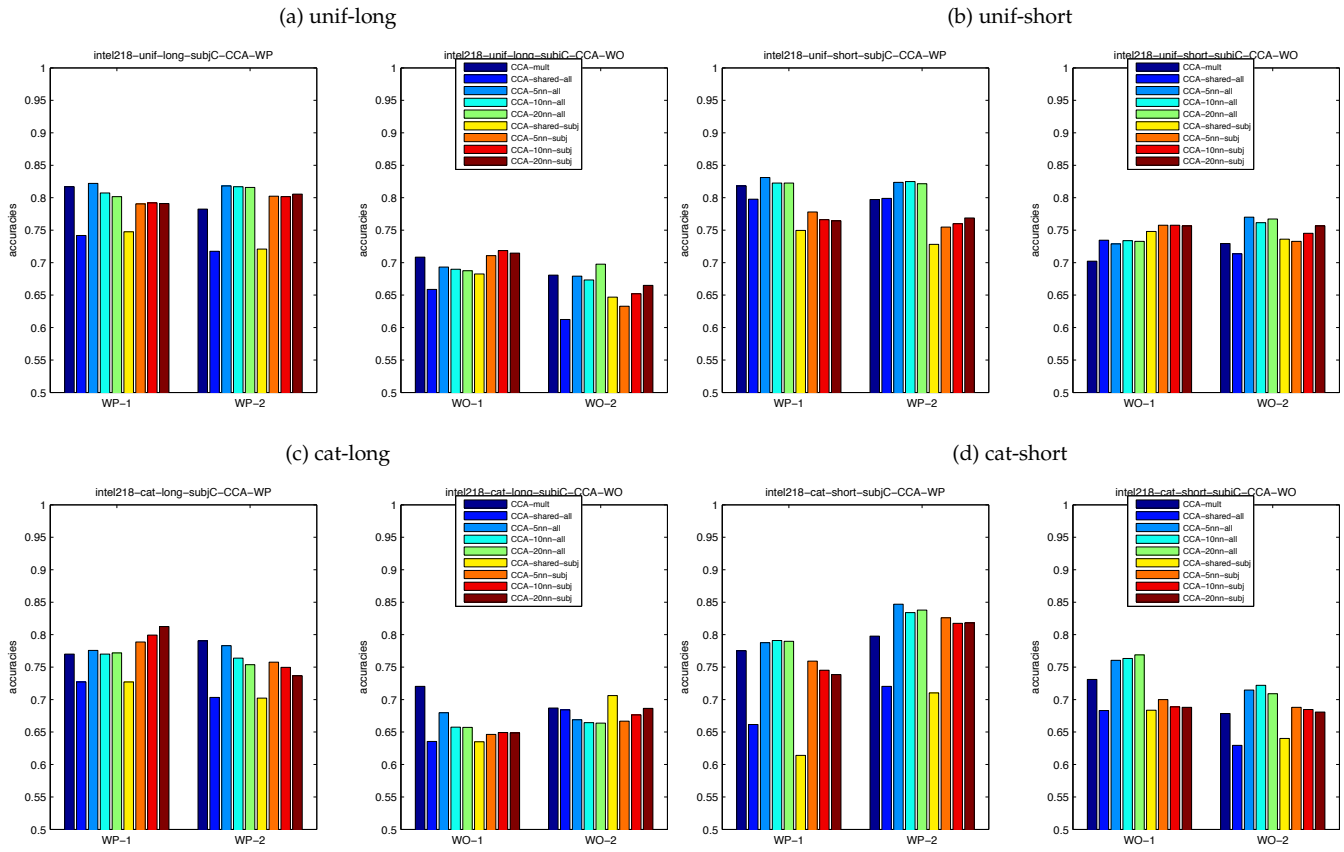


Figure 4.3.10: Accuracies for subject C when using all the subjects in both studies vs when we have only subject C in each study, using the intel218 features and leaving out words based on the available sets.

- intel218-cat-short (WP-2, WO-1)

And for subject C, in two cases out of 16, the subj accuracies outperform the all accuracies:

- intel218-unif-short (WO-1)
- intel218-cat-long (WP-1)

while in 11 out of 16 cases, the subj accuracies underperform the all accuracies.

- intel218-unif-long (WP-1, WO-2)
- intel218-unif-short (WP-1, WP-2, WO-2)
- intel218-cat-long (WP-2, WO-1)
- intel218-cat-short (WP-1, WP-2, WO-1, WO-2)

To summarize, for subject A (9 out of 16 cases showing no significant effect), there does not appear to be any significant effect in incorporating data from the other subjects to perform the imputation, but for subjects B (8 out of 16 cases showing benefits) and C (11 out of 16 cases showing benefits), there are clear benefits when we use data from the other subjects to perform the imputation. Coincidentally, we see that the accuracies obtained when we predict subject A's data are the highest among the three subjects.

These figures also show the effect of the imputation method versus the effect when we combine data from all subjects only from the same study (the accuracies labeled as CCA-mult). In the following cases, there is one or more methods based on imputation yielding significantly better accuracies compared to the accuracies of the CCA-mult method:

- subject A: unif-long (WP-1, WP-2, WO-1), unif-short (WP-1), cat-short (WP-1, WP-2)
- subject B: unif-long (WP-2, WO-1), unif-short (WP-1, WP-2, WO-1), cat-long (WP-1, WP-2), cat-short (WP-1, WP-2, WO-2)
- subject C: unif-long (WP-2, WO-1, WO-2), unif-short (WP-2, WO-1, WO-2), cat-long (WP-1), cat-short (WP-1, WP-2, WO-1, WO-2)

and in the following cases, the accuracies of CCA-mult method outperform the accuracies of all the imputation-based methods.

- subject A: unif-long (WO-2), cat-long (WO-1, WO-2), cat-short (WO-1)
- subject B: unif-long (WO-2)
- subject C: cat-long (WO-1)

Here we see that although when we consider mean accuracies, there are no significant differences in general between the accuracies of the imputation based methods versus the accuracies of the CCA-mult method, when we consider accuracies on a per-subject basis, we see more variations, and in particular, for two of the three subjects that we consider (subjects B and C), there are significant improvements (10 out of 16 cases for subject B, and 11 out of 16 cases for subject C) when we perform imputation compared to when we use the CCA-mult method.

4.3.5.2 Investigating the nearest neighbors

We now dig deeper into what the imputation method does. We first focus on the cases when we combine data of all the subjects in both studies and use 5 nearest neighbors. First, we look into the accuracy of the imputed values compared to the actual values. In particular, for each imputed feature, we calculate the difference between the imputed value and the held-out actual value. Figure 4.3.11 shows the distributions of these differences in four cases:

- unif-long-2, subject 1WP
- unif-long-2, subject 11WO
- cat-long-2, subject 1WP
- cat-long-2, subject 11WO

These four cases give a good representation of all the cases we have when the data from all the subjects in both studies are used. First, we see from figure 4.3.11 that in most cases, the averages of the difference between the imputed values and the actual values are relatively close to zero. So in general, the imputation method is relatively unbiased. On the other hand, from figure 4.3.11, we also see that in majority of the cases, the whiskers of a particular boxplot extends to beyond ± 1. This is accompanied by the relatively heavy presence of outliers. In order to see the significance of these characteristics, in figure 4.3.12, we show in the form of boxplots the distribution of the values to be imputed for each word in the same four cases. Looking at the two figures, it is hard to tell any significant differences between the plots in figure 4.3.11 and the plots in figure 4.3.12. In other words, the range of differences between the imputed and the actual values are in line with the range of the actual values themselves.

Also, note that although we do not show boxplots for these cases, these characteristics also hold when we leave out words from the "short" lists. We do see improved accuracies when we use the "short" lists. Based on the difference between the imputed values vs the actual values, the improved accuracies in the "short"-list cases seem to be due mostly to the availability of actual instances instead of the imputation method.

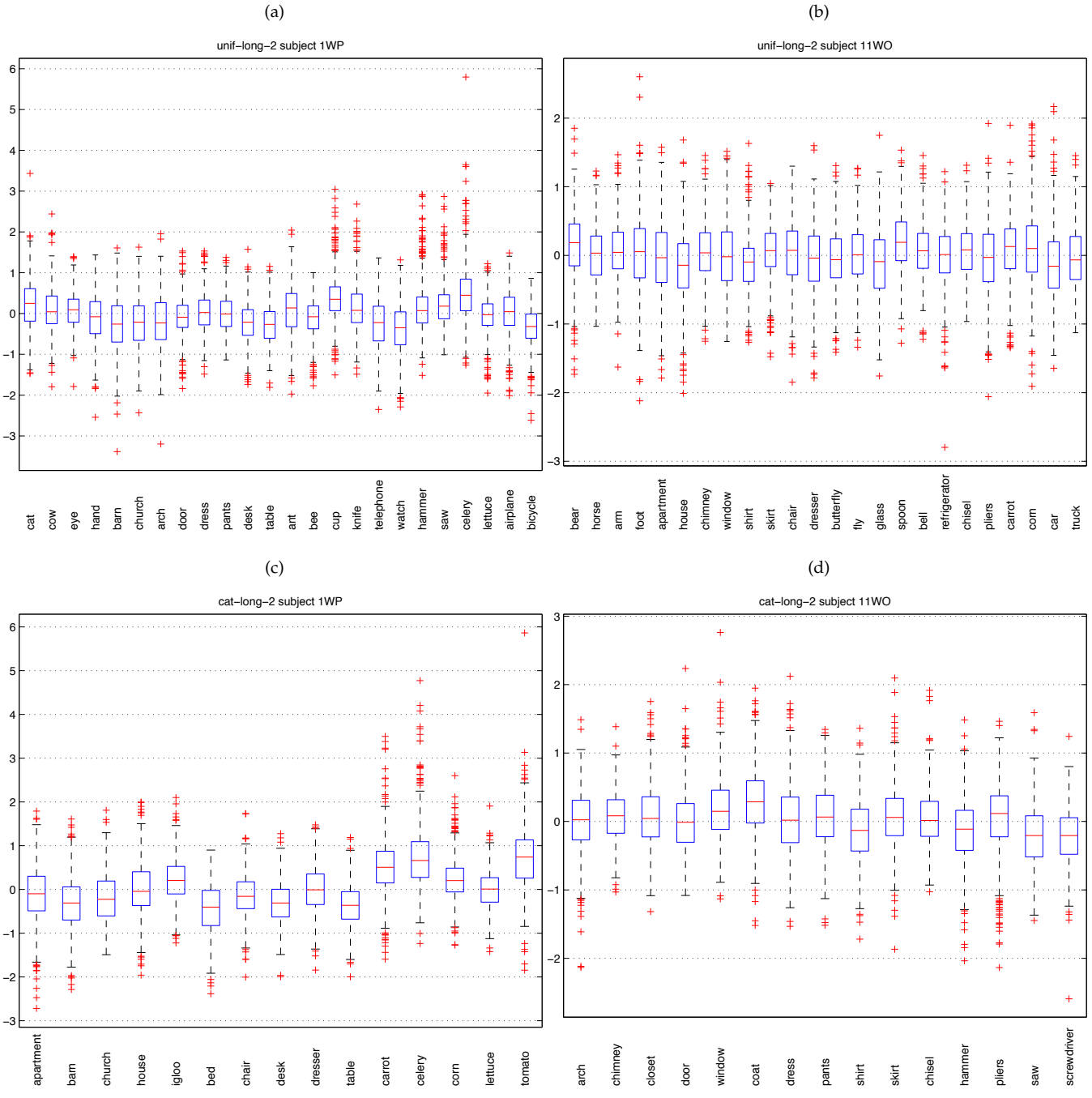


Figure 4.3.11: Distributions of the differences between actual and imputed values in four cases. The following is based on MATLAB's documentation for the `boxplot` command. For each word, the central mark in the box represents the median of the differences and the edges of the box represent the 25th and 75th percentiles of the differences. The whiskers extend to the most extreme data points not considered outliers. Points are considered outliers if they are larger than $q_3 + 1.5(q_3 - q_1)$ or smaller than $(q_1) - 1.5(q_3 - q_1)$, where q_3 and q_1 are the 25th and 75th percentiles of the differences, respectively.

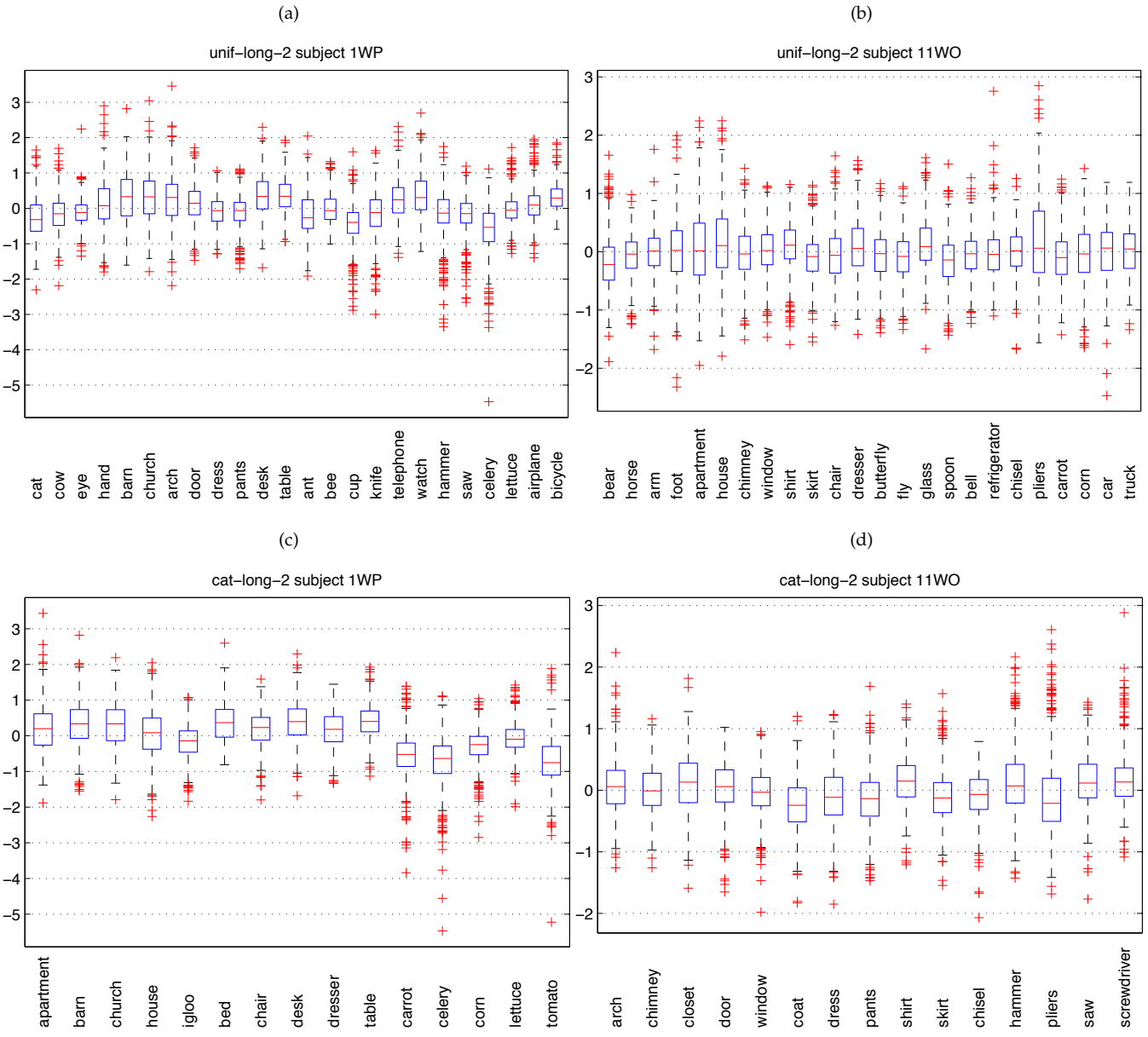


Figure 4.3.12: Distributions of the actual values in four cases.

We now investigate the sources for the imputed values. First, we look into the subject sources. Figure 4.3.13 shows each subject contribution counts as heat maps, for four cases: unif-long-1, unif-short-2, cat-long-1, and cat-short-2. For each map, each point on the horizontal axis corresponds to a subject source, i.e. the subject from which the relevant contribution counts are drawn, and on the vertical axis, each point corresponds to a subject whose features we are imputing, or the subject target. The color at each element indicates the contribution counts from the subject source to imputing data in the subject target, ranging from blue (zero count) to dark red (max counts).

In the "unif" cases, when we are imputing the features for the WP subjects, we see relatively uniform contributions from all the WO subjects. On the other hand, when we are imputing the features for the WO subjects, there are more contributions from subjects 4WP and 7WP relative to the contributions from the other subjects, while there are fewer contributions from subject 1WP compared to the contributions from the other subjects. The latter fact is somewhat interesting, in light of the fact that subject 1WP consistently exhibits the highest individual accuracies across various base features. We see somewhat similar trends in the "cat" cases, although in the cat-long-1 case, we see relatively more contributions to the WO subjects from subject 1WP compared with the contributions in the other three cases.

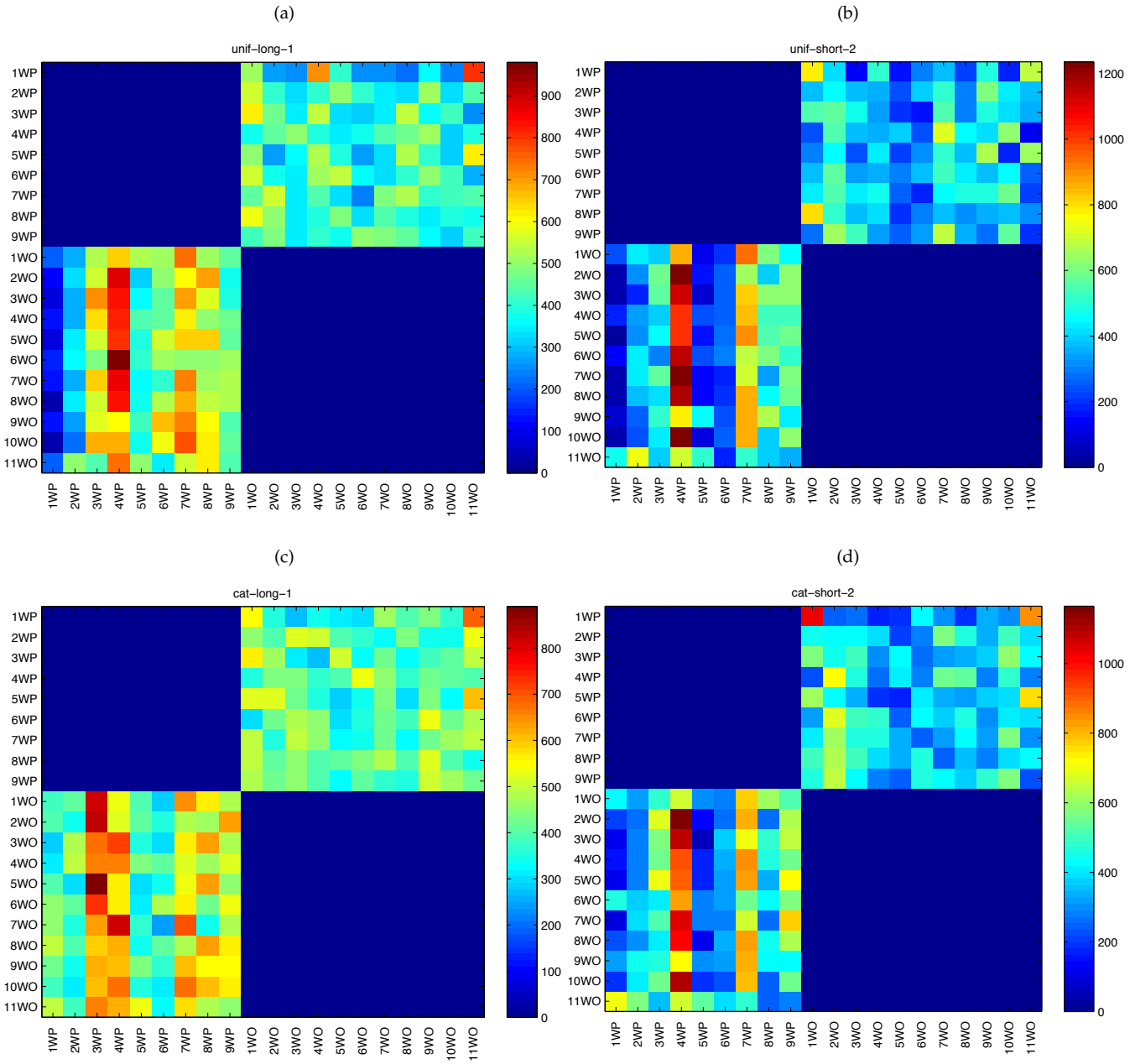


Figure 4.3.13: Heat maps of the subject sources and destinations for imputed values. An entry in the horizontal axis denotes a particular subject source, i.e. the subject which provides contribution for imputation of values in the subject targets, shown in the vertical axis. The color at each entry reflects the number of the contribution from the subject source corresponding to that entry to the subject target corresponding to that entry.

In addition, we can look into how the location of a feature used to impute corresponds to the location of the imputed feature. In particular, we consider we consider locations in terms of regions of interest (ROIs). Figure 4.3.14 shows the heat maps of the counts of ROI contributions in four cases:

- unif-long-1 subject 1WP
- unif-long-2 subject 1WO
- cat-long-2 subject 1WP
- cat-long-1 subject 1WO

In each map, each point in the horizontal axis represents the ROI of a feature that contributes to the imputation, and in the vertical axis each point represents the ROI of a feature whose value we want to impute. First, we see that there are fewer ROIs to impute in the WP cases compared to the ROIs in the WO cases. In particular, a majority of the ROIs in the WP cases are located in the posterior part of the brain, shown in the middle of the ROIs listed. This is an effect of the stable voxels chosen for these experiments, which are concentrated in the posterior ROIs for the WP cases but are more spread out throughout the brain in the WO cases. Looking at the entries, we see that the contributions to these posterior ROIs in the WP cases (both ”unif” and ”cat”) come mostly from the posterior ROIs of the WO subjects. In the WO cases, although there are more ROIs to impute, we also see that the posterior ROIs in the WO cases get most of their contributions from the posterior ROIs of the WP subjects. The other ROIs present in the WO cases get their contributions relatively uniformly from the available ROIs of the WO subjects.

Also of note is the presence of features not associated with any ROIs, labeled as NOROIS on the maps. As can be seen on the maps, there might be significant contributions both from and to these features.

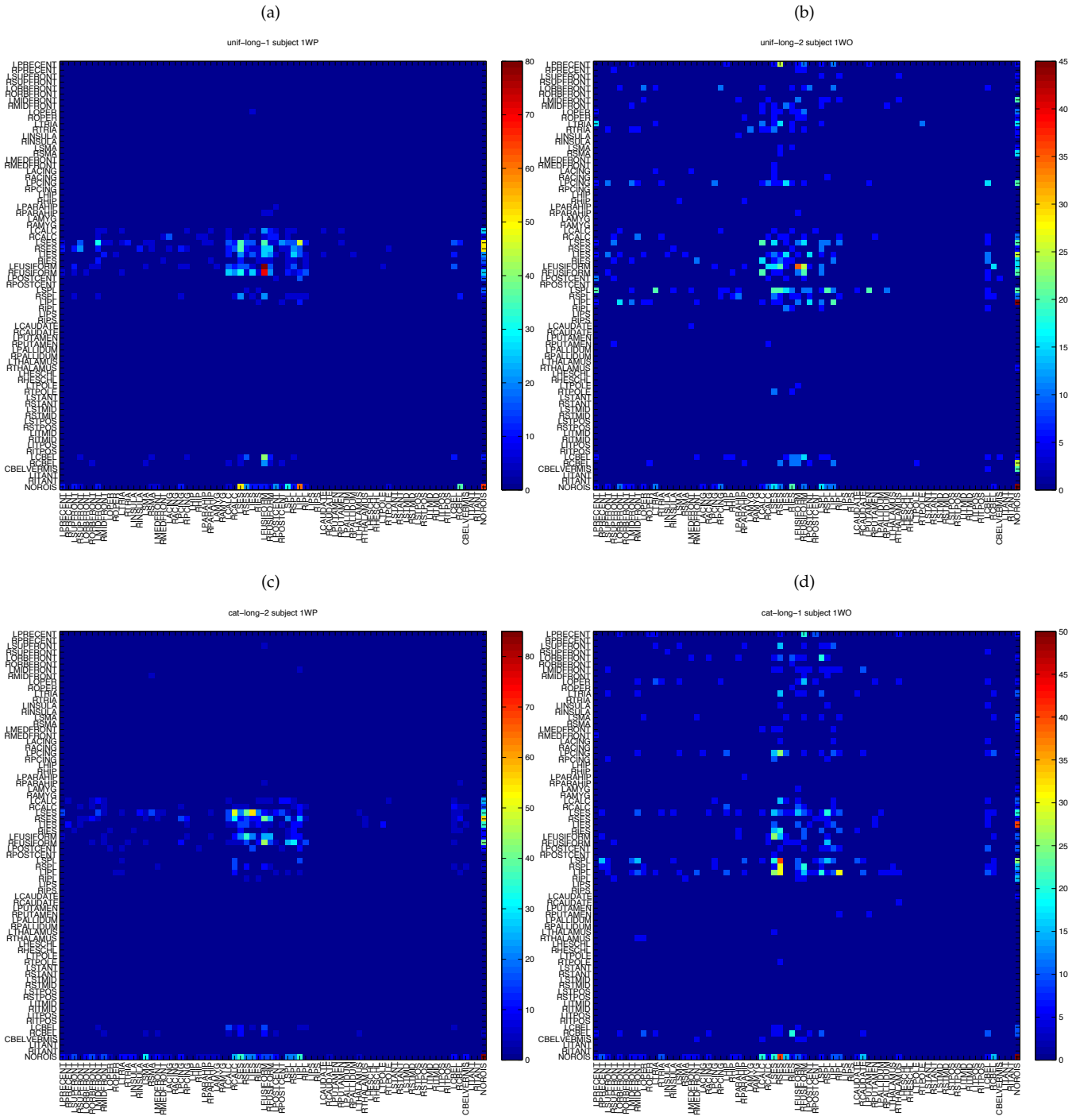


Figure 4.3.14: Heat maps of the ROI sources and destinations for imputed values. A glossary of the ROI terms is provided in appendix C. The horizontal axis represents ROI sources, i.e. ROIs that provide contribution for the imputation of values in the ROI targets, shown in the vertical axis. Each entry denotes with color the number of contributions of the entry's ROI source to the entry's ROI target.

We perform similar analyses for when we apply the model taking one subject from each dataset. We first look at the differences between the imputed values vs the actual values, shown in the form of boxplots in figure 4.3.15. We select four cases to show:

- unif-long-2 for subject 1WP
- unif-long-2 for subject 1WO
- cat-long-2 for subject 1WP
- cat-long-2 for subject 1WO

Comparing figure 4.3.15 and figure 4.3.11, we do not notice any significant differences in terms of the distributions of the differences. In both cases, the means of the distributions in most cases are close to zero, and the spread and outlier degrees are comparable.

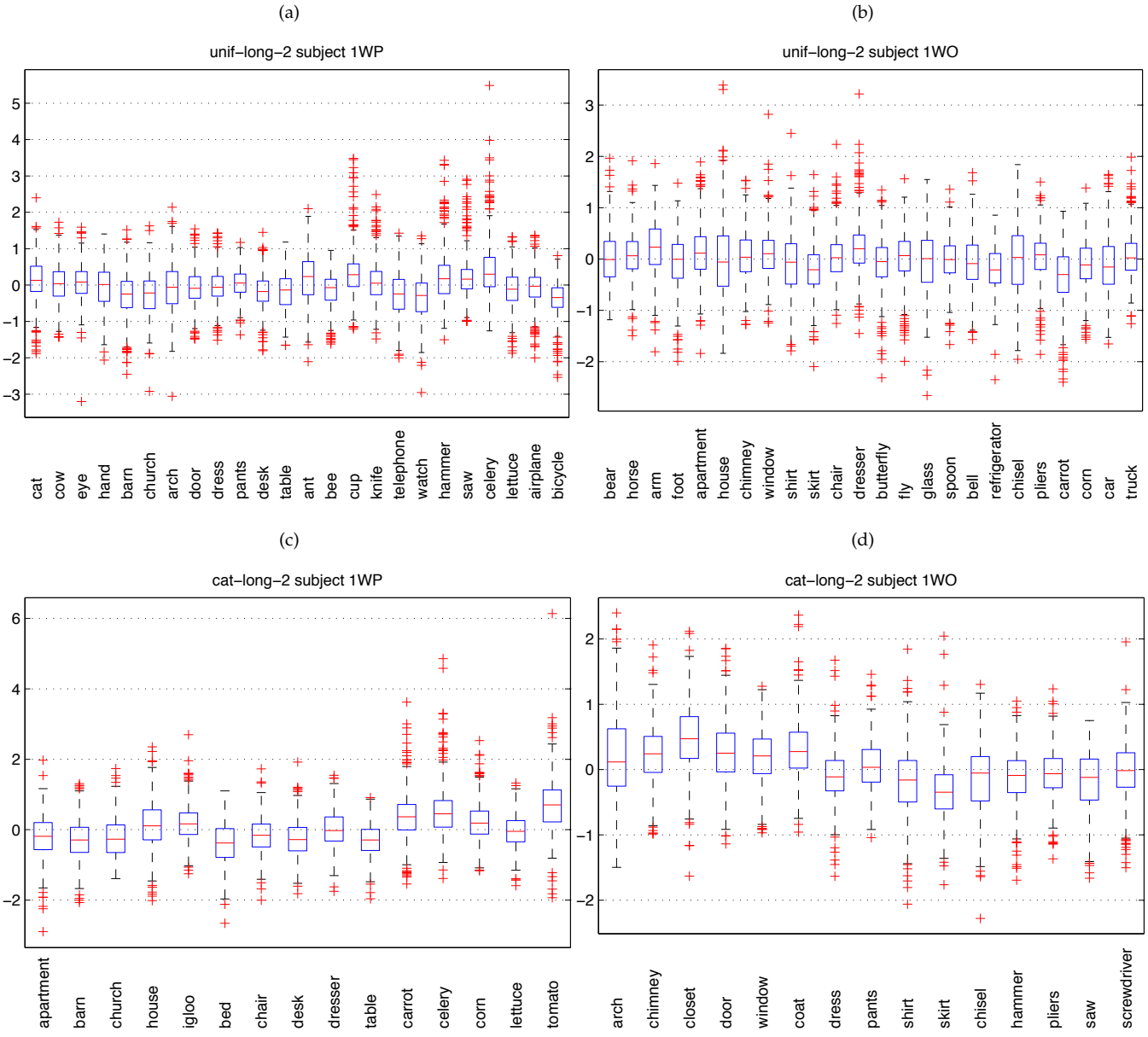


Figure 4.3.15: Distributions of the differences between actual and imputed values in four cases involving subject A.

We can also look at compare the location of the ROI of a feature used to impute with the location of the ROI of a feature to be imputed, shown in figure 4.3.16. Again, like in figure 4.3.14, we see that the posterior ROIs of the WO subject provide most of the contributions to the posterior ROIs of the WP subject, and vice versa.

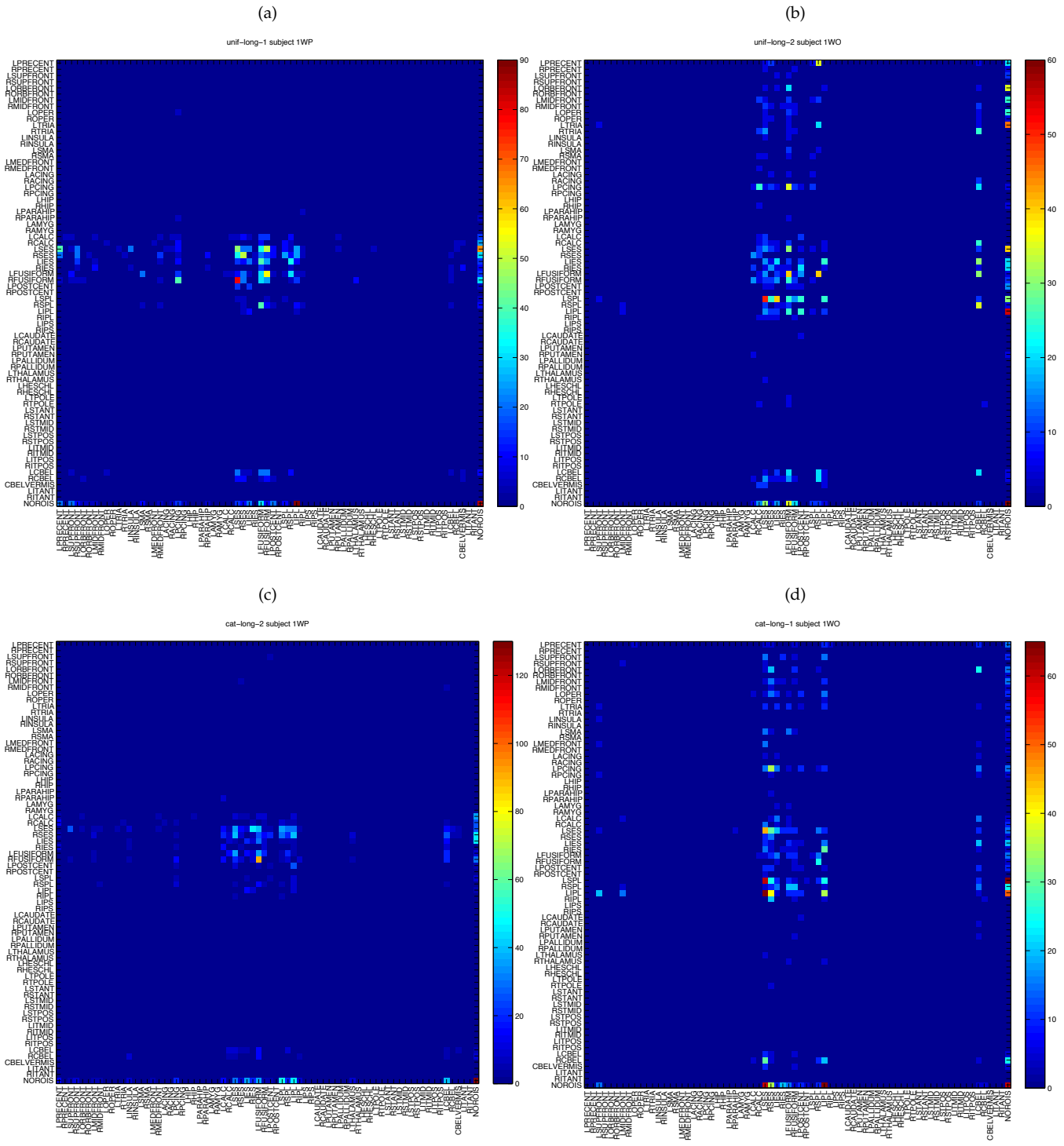


Figure 4.3.16: Heat maps of the ROI sources and destinations for imputed values when we do analysis on subject A.

4.3.6 Discussion

1. How does leaving the words affect the accuracies of the baseline model?

   When we leave out words from the training examples, the accuracies of the baseline model invariably decline. It is not surprising that a greater decline is incurred when we leave out more words, and when we leave whole categories out compared to when we leave out words uniformly from each category.

2. Does the imputation scheme yield better accuracies compared to methods relying on only the shared instances?

   Yes, the imputation scheme enables us to go beyond using only the shared instances and obtain better accuracies as a result. We see both when considering the data from all the subjects in both studies (as shown in figures 4.3.1, 4.3.2, 4.3.3, and 4.3.4), and when considering specific subjects (as shown in figures 4.3.5, 4.3.6, and 4.3.7).

3. Does the imputation scheme yield better accuracies compared to the baseline model and the within-study common-factor approaches?

   Compared to the baseline model's accuracies, we manage to obtain better accuracies using the imputation scheme in conjunction with CCA in a number of cases. However, in cases where we have more than one subject from each study, it is not clear that the imputation scheme yields significantly better accuracies compared to within-study approaches. Nonetheless, when there is only one subject in each study, with the within-study common-factor approaches ruled out, our results involving each of the three common subjects indicate that there is still value provided by the imputation schemes.

4. How do the accuracies vary with the number of nearest neighbors used?

   Based on the numbers of nearest neighbors used in the experiments, there is no clear indication that the accuracies significantly depend on the numbers of nearest neighbors.

5. How do the imputed values compare to the actual values?

   On average, the difference between the imputed and the actual values are close to zero, and the variance is comparable to the variance of the actual values.

6. What is the distribution of the sources for the imputed values (in terms of subjects and locations)?

   There is some evidence that a good number of the sources for the imputed values come from similar locations as the destinations in terms of the brain regions. On the other hand, in the case of experiments involving multiple subjects in each study, when imputing the values for the WP subjects, the contributions come somewhat uniformly from all the WO subjects, while when imputing the values for the WO subjects, the contributions are relatively concentrated from one or a few subjects.

4.4 Summary

In this chapter, we have seen the application of the linear factor analysis framework described in chapter 3 to predictive analysis of real fMRI data. The results in this chapter show that by learning common factors of the fMRI data across subjects and studies, we can improve the accuracies of the predictive task considered here. In particular, we see most of improvements when we use canonical correlation analysis (CCA) to learn the common factors, while when we use principal components analysis (PCA) to learn the common factors, we do not find significant improvements over not integrating information across subjects and studies.

One limitation of the linear factor analysis framework is that it requires that all the datasets to be analyzed jointly have matching instances, i.e. for each instance in a particular dataset, we need to be able to find a corresponding instance in all the other datasets. In chapter 3, we describe an imputation method to get around this limitation, and in this chapter we have shown results when applying this imputation method. The results show that the method has potential especially when each dataset contains some instances that are not present in any of the other datasets. The imputation method is still limited in the sense that it needs

the datasets to have at least some shared instances. The approach outlined in the next chapter has the potential to bring about another alternative method to deal with datasets with non-matching instances, and this method will be outlined in chapter 6 as a direction for future works.

Chapter 5

Case Study 2

Abstract

This chapter describes the second case study of applying the linear factor analysis framework in predictive fMRI data analysis context. The application is the same as that in case study 1, i.e. the task of predicting the fMRI activations for arbitrary concrete nouns. However, in this case study, we investigate learning factors common to both fMRI data and the semantic features. The experiments in this case study show that the accuracies obtained when we learn factors common across both fMRI data and semantic features are comparable to the accuracies obtained when we learn factors common across the fMRI data only (the latter accuracies are obtained in case study 1). However, the accuracies show steeper declines as the number of factors increases.

This case study concerns the same predictive task considered in the first case study, i.e. the task of predicting the fMRI activations for arbitrary concrete objects. However, in this second case study, we ask the question whether finding some common latent dimension shared by both the predefined semantic features and the fMRI data can lead to better prediction. We explore this question by applying some of the factor analytic methods described in chapter 3 to both the predefined semantic features and the fMRI data to discover dimensions common between the predefined semantic features and the fMRI data, and investigate how well the dimensions perform when applied to the predictive task. Because we involve fMRI data from multiple subjects and/or studies, these dimensions will also be common across these subjects and/or studies. We run experiments on the WP dataset (section 1.3.3) as well as the WO dataset (section 1.3.4).

5.1 Method

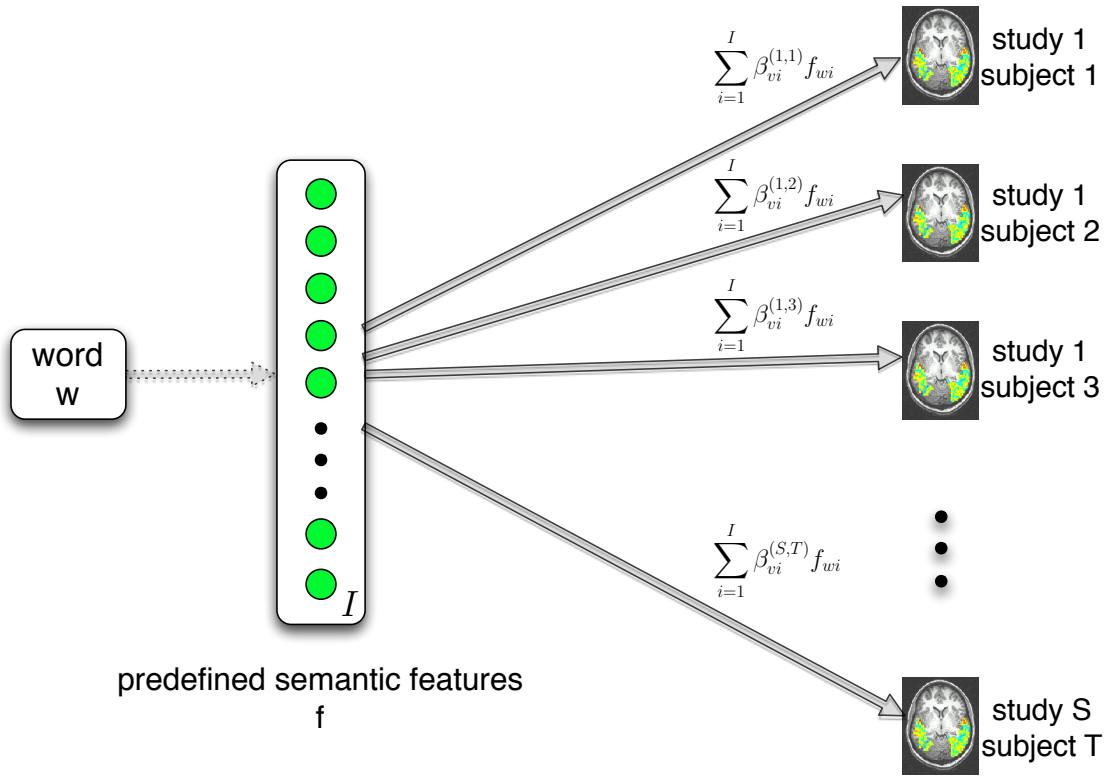


Figure 5.1.1: The baseline model of Mitchell et al. (2008), expanded to take into account the potential presence of fMRI data from multiple subjects and/or studies, with semantic features denoted as predefined semantic features.

We again take as the baseline model the model of Mitchell et al. (2008), which we show again in figure 5.1.1. Here we relabel the *base features* as *predefined semantic features*, to distinguish from the *latent features/factors* that are going to be learned. In this case study, we modify the baseline model as follows. Instead of assuming the direct mapping from the predefined semantic features to the fMRI activations as shown in figure 5.1.1, we now assume the existence of a few latent factors that generate both the predefined semantic features and the fMRI activations, shown as latent factors z in figure 5.1.2. In turn, both the predefined semantic features and each subject-study's fMRI data are generated as linear combinations of the latent factors z. As formulated, the model can be written in the style of equation (3.2), where we can consider the predefined semantic features as one dataset and each subject-study's fMRI data as an additional

dataset. Mapping the model to equation (3.2), we can take the predefined semantic features to be the first dataset, so for a specific word w, the vector $\mathbf{y}_w^{(1)}$ (replacing i with w) contains the predefined feature values for w, and for each fMRI dataset, assigned to the index m (ranging from 2 to M), the vector $\mathbf{y}_w^{(m)}$ contains the corresponding fMRI activations for word w. Then the latent factors for word w will be the vector $\mathbf{z}_w$, and the coefficients for the linear combination for each dataset m will be contained in the matrix $\mathbf{W}^{(m)}$.

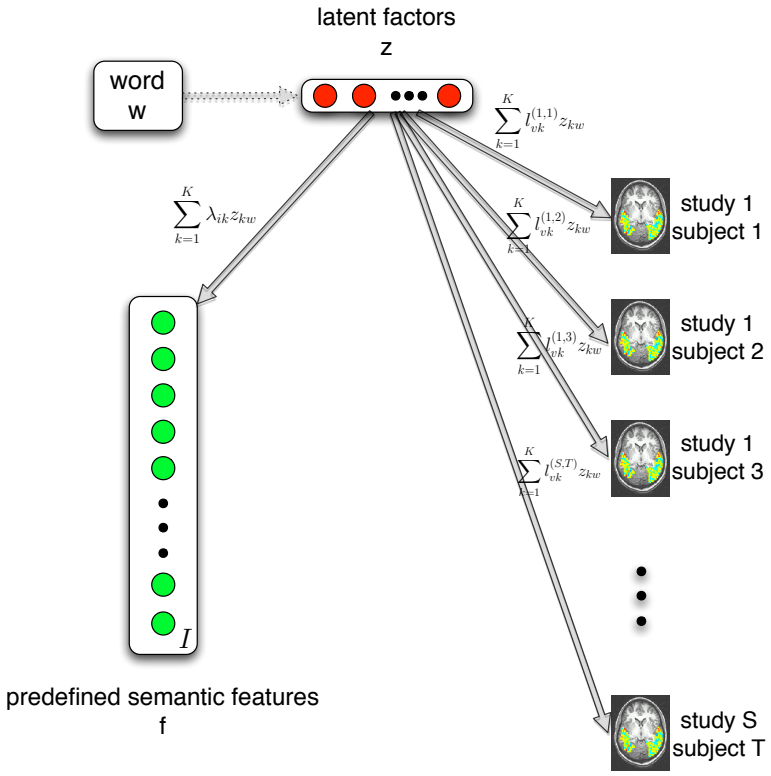


Figure 5.1.2: The latent-factor augmentation to the predictive model of Mitchell et al. (2008)

The correspondence between the model shown in figure 5.1.2 and equation (3.2) means that we can use any of the methods described in chapter 3 to estimate the parameters of the model. In particular, we apply the following methods:

- **CCA-concat**, the classical 2-way CCA, where we concatenate all the fMRI data into one data matrix, with the predefined semantic features forming the other data matrix.
- **CCA-mult**, the multiple-dataset CCA, treating the fMRI data from each subject separately from the data from the other subjects and the predefined semantic features; in other words, we have one data matrix for each subject, along with one data matrix for the predefined semantic features.
- **PCA**, where we combine both the fMRI data and the predefined semantic features into one data matrix.

Similar to the baseline model of figure 5.1.1, once trained, the latent-factor model of figure 5.1.2 can be used to predict the fMRI activations for words that are not used to train the model. Predictive accuracies will be the principal way for us to evaluate the model. As in the baseline model, the way to generate predictions for unseen words is to take the predefined semantic features for those words and use the parameters of the model to generate the predicted fMRI activations. In the latent-factor model, before we can obtain the predicted fMRI activations, we first have to derive the latent factors z associated with the unseen words. Here are some details on how we do this for each of our three methods:

- **CCA-concat**, we multiply the learned CCA loadings for the predefined semantic features with the predefined feature values for the unseen words.
- **CCA-mult**, identical to the **CCA-concat** case.
- **PCA**, we take the PCA loading matrix for the predefined semantic features and compute its pseudo-inverse; we then left-multiply the predefined semantic feature values for the unseen words with this pseudo-inverse.

After we derive the latent factor values for the unseen words, we then need to project these values to the fMRI activations. Again, here are some details on how we do this for each of our three methods:

- **CCA-concat**, we multiply the pseudo-inverse of the learned CCA loadings for the fMRI data with the derived latent factor values for the unseen words, and then we extract the subset of the results corresponding to the fMRI dataset of interest.
- **CCA-mult**, we multiply the pseudo-inverse of the learned CCA loadings for the fMRI dataset of interest with the derived latent factor values for the unseen words.
- **PCA**, we multiply the subset of the PCA coefficients corresponding to the fMRI dataset of interest with the derived latent factor values for the unseen words.

For the CCA-mult and PCA methods, the procedures are essentially the same as the procedures employed for the corresponding methods in case study 1. Next, we describe the experiments performed.

5.2 Experiments

The experiments performed in this case study are intended to obtain answers to the following questions:

1. Can we get better prediction, as measured by prediction accuracies, using the latent-factor model as opposed to the baseline model, for different kinds of predefined semantic features?
2. How do the three methods used to estimate the parameters of the latent-factor model compare to one another?
3. How do the prediction accuracies vary with the number of components?
4. How would the prediction accuracies be affected if we use fMRI data from multiple studies as opposed to fMRI data from a single study?
5. How do the prediction accuracies compare with different kinds of predefined semantic features?
6. What kind of semantic information, if any, is contained within each component? How does this information vary if we use data from multiple studies, and if we use different kinds of predefined semantic features?
7. How is the information contained in a component reflected in the brain for each subject?
8. How do the accuracies compare with those in case study 1?

Also note that for **CCA-concat** and **CCA-mult**, we set the regularization parameter $\kappa = 0.5$ in advance. This is the setting used in the previous chapter, and we do not explore whether this setting is optimal.

5.2.1 fMRI Datasets

The WP and WO datasets described in sections 1.3.3 and 1.3.4 are used. We refer to these sections for descriptions about how the data were pre-processed.

5.2.2 Predefined semantic features

We use the 485verb and intel218 features as used in case study 1 and described in section 4.2.2.

Pre-processing for the predefined semantic features As in case study 1, we adjust the scaling of the 485verb features by normalizing them so that each instance has length one. This is done for all the methods run in our experiments. In addition, for the factor analytic methods, we normalize the intel218 features so that for each instance it has length one. Moreover, for the fMRI data we do the following normalization:

- **CCA-concat** We normalize the concatenated fMRI data matrix so that the vector for each instance (in this case the vector corresponds to several fMRI images concatenated together) has length one.
- **CCA-mult** We normalize each dataset separately so that in each data matrix, the vector for each instance (in this case the vector corresponds to an fMRI image) has length one.
- **PCA** We normalize each fMRI dataset separately so that in each dataset, the vector for each instance has length one, and then we concatenate all the datasets to form one data matrix.

The normalization is performed as a way to have the datasets to be combined to be of similar scale. We do not need to perform this in case study 1, since we are deriving learned factors from only fMRI datasets, which have similar scales already. But in this case study, the semantic features in their original forms can potentially have scales different from the fMRI datasets.

5.2.3 Evaluation

We use the evaluation procedure described in section 4.2.3, including the jackknife procedure to estimate confidence intervals for the accuracies described in that section.

5.3 Results

5.3.1 Accuracies

Figure 5.3.1 shows the mean accuracies of all of the subjects in each study of the various methods applied in this case study. The top row shows the results using the 485verb features, while the bottom row shows the results using the intel218 features. In the figure, we show the accuracies as a function of the number of components used, where we consider 1, 2, 3, 4, 5, 10, 20, 30, 40, and 50 components. We also show the accuracies for the baseline LR method in the leftmost vertical bar in each group.

The figure shows that the accuracies for all the factor analytic methods considered increase as the number of components used increases until they reach some peaks, and then the accuracies plateau or decrease as we add more components. The peak accuracies are obtained when we use around five components, with the exception of the CCA-mult method for the WP dataset in conjunction with the 485verb features, in which case the peak is reached only when 20 components are used. These figures show that for each of the factor analytic methods considered, the first few components contain the information relevant for prediction.

The figure also shows that among the factor analytic methods being considered, the best accuracies are obtained using the CCA-mult method, while the worst accuracies are obtained using the PCA method. These suggest that there is some value in considering each dataset as being distinct, in contrast with concatenating the datasets as is done in the PCA method and to some extent in the CCA-concat method. The results, especially in the 485verb case, also show that there is some value in jointly analyzing the WP and WO datasets; in the intel218 case, the accuracies obtained from combining the datasets together are not significantly different from those obtained when each dataset is considered separately. With the exception of the intel218-WO case, the peak accuracies of the CCA-mult method are also significantly better compared to the accuracies of the baseline LR method; in the intel218-WO case, the peak accuracies of the CCA-mult method are only marginally better compared to the baseline LR accuracies.

Next we consider the information present in the components extracted using the various factor analytic methods. Note that for each component/factor, the scores and loadings are invariant to reversing the signs of both simultaneously.

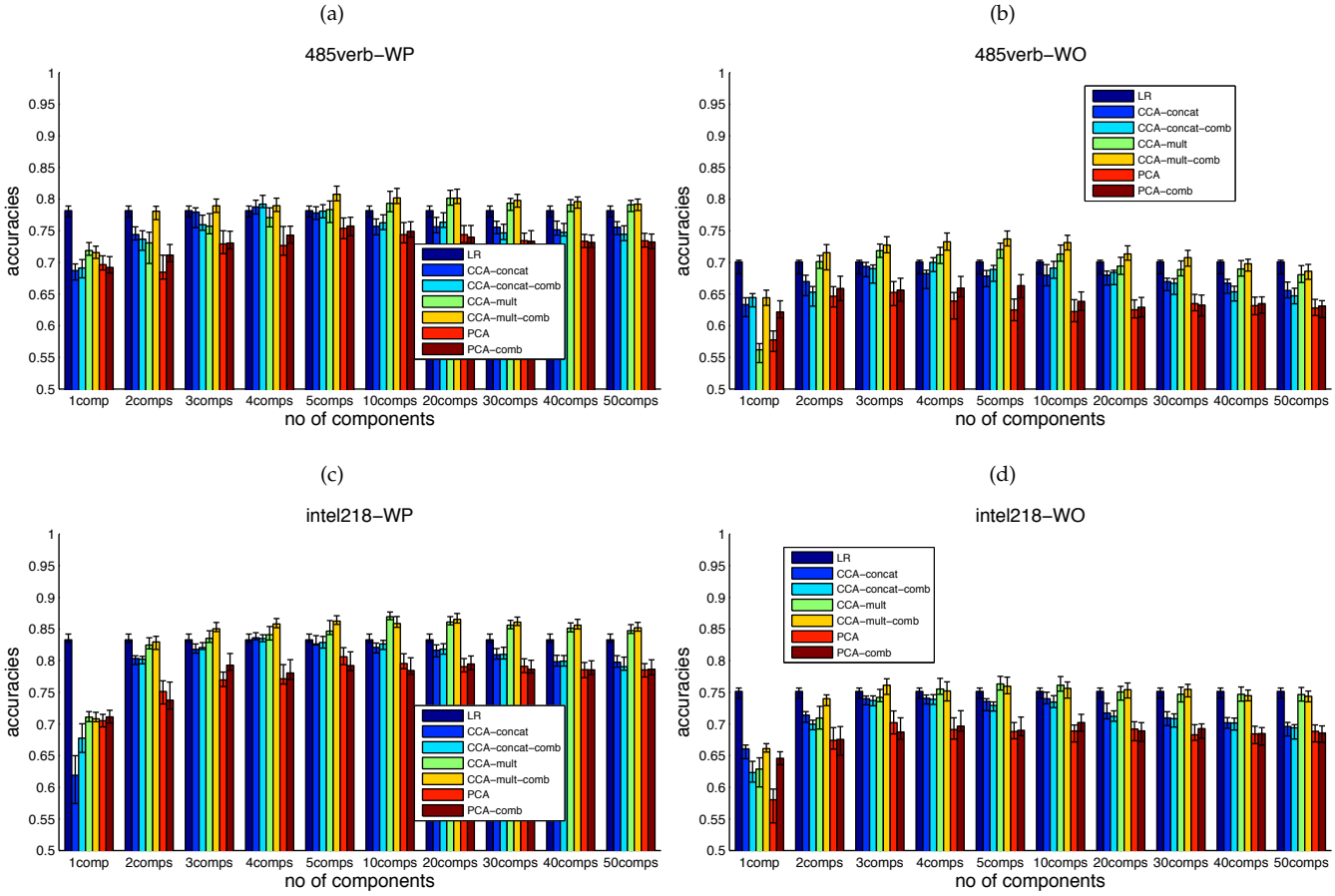


Figure 5.3.1: Mean accuracies of the baseline model (LR) along with those of the implementations of the latent-factor model. Note that unlike what is done in chapter 4, here we show the accuracies for the LR method as bars in all the bar chart groups.

5.3.2 Word scores

We present tables of the stimulus words with the most positive and most negative scores in each of the factor analytic methods considered in this case study.

CCA-concat Tables 5.3.1 and 5.3.2 show the top stimulus words extracted by the first five components using the CCA-concat method for the 485verb and intel218 features, respectively. We first look at the variations across datasets given a particular set of features. Let us consider the 485verb case. For the WP dataset, on one side, the first component seems to represent the shelter dimension, while the other side contains three tool words along with the words "celery" and "carrot". The tool dimension is also present in the first component for the WO and the WP-WO datasets, but the shelter dimension is not apparent in these two datasets. On the other hand, the second component of the WO dataset exhibits the short-word-length dimension, i.e. on one side there is a grouping for short words. The short-word-length dimension seems to still be present for the WP-WO dataset in the second learned component. Other interesting groupings include the apparel dimension (component 4 of WP, component 5 of WO, component 3 of WP-WO) and the body-part dimension (component 3 of WP).

Now we consider the intel218 case. In the first component of the WP dataset, we see the shelter dimension on one side and the dimension of short-word animals on the other side. The shelter dimension is also present in the first component of the WO and the WP-WO datasets, while the animal dimension is present in the first component of the WP-WO dataset only. For all three datasets, the second component contains a dimension for four-legged animals on one side, and tools on the other side, while the third component contains the body-part and vegetable dimensions.

Comparing these results for both 485verb and intel218 cases, we see that the intel218 features induce relatively more stable (across fMRI studies) groupings in the first three components in the three datasets compared to the groupings obtained in the 485verb case. More generally, it appears that the features (485verb and intel218) play a major role in influencing the components produced by the CCA-concat method. The groupings shown in these tables therefore might have a lot to do with the groupings implicitly present within each set of features, either 485verb or intel218.

		component 1	component 2	component 3	component 4	component 5
WP	positive	celery	door	foot	skirt	ant
		knife	window	arm	shirt	table
		pliers	refrigerator	leg	pants	arm
		carrot	leg	truck	dress	igloo
		screwdriver	pants	hand	cat	pants
	negative	train	beetle	lettuce	screwdriver	horse
		apartment	dog	cup	celery	chisel
		church	bear	igloo	pliers	airplane
		barn	fly	cow	carrot	train
		house	airplane	house	knife	pliers
WO	positive	pliers	cat	dog	cup	pants
		celery	eye	carrot	door	skirt
		carrot	leg	igloo	cow	shirt
		knife	bee	train	house	dress
		screwdriver	dog	lettuce	window	beetle
	negative	car	screwdriver	leg	bicycle	key
		desk	refrigerator	door	hand	ant
		house	chisel	arm	foot	arch
		church	pliers	window	train	igloo
		train	bicycle	pants	truck	saw
WP-WO	positive	celery	door	pants	truck	butterfly
		pliers	window	leg	airplane	horse
		knife	refrigerator	skirt	train	tomato
		carrot	dresser	foot	bicycle	dress
		screwdriver	screwdriver	shirt	arm	bicycle
	negative	train	cat	chimney	cup	ant
		apartment	dog	celery	door	arm
		church	bee	screwdriver	window	saw
		car	bear	carrot	cow	igloo
		desk	fly	pliers	house	arch

Table 5.3.1: Stimulus word rankings of the first five components learned by the CCA-concat method when we use the 485verb features.

		component 1	component 2	component 3	component 4	component 5
WP	positive	ant	bear	leg	tomato	bear
		cat	horse	foot	butterfly	dog
		fly	cow	hand	igloo	cat
		dog	airplane	arm	lettuce	cow
		bee	dog	pants	shirt	door
	negative	apartment	knife	celery	saw	hand
		church	spoon	tomato	bicycle	arm
		closet	screwdriver	carrot	hammer	truck
		barn	glass	lettuce	truck	foot
		house	chisel	corn	screwdriver	car
WO	positive	apartment	dog	foot	celery	pants
		train	bear	leg	lettuce	coat
		house	cat	arm	tomato	skirt
		airplane	cow	hand	corn	dress
		church	horse	pants	carrot	shirt
	negative	carrot	screwdriver	tomato	saw	leg
		hand	pliers	corn	hammer	chimney
		celery	knife	carrot	chisel	arm
		corn	chisel	igloo	bicycle	ant
		lettuce	spoon	celery	pliers	foot
WP-WO	positive	ant	bear	leg	saw	hand
		bee	horse	foot	hammer	arm
		cat	cow	arm	bicycle	foot
		carrot	dog	hand	chisel	carrot
		fly	cat	pants	screwdriver	car
	negative	apartment	knife	celery	tomato	bear
		church	screwdriver	carrot	lettuce	cat
		house	spoon	tomato	corn	dog
		train	pliers	corn	igloo	dress
		closet	chisel	igloo	celery	coat

Table 5.3.2: Stimulus word rankings of the first five components learned by the CCA-concat method when we use the intel218 features.

CCA-mult Now we consider the top stimulus words extracted using the CCA-mult method, shown in tables 5.3.3 and 5.3.4. In light of the discussion for the CCA-concat method, when we compare the two tables, the words contained in them are very similar across the two tables. This means using the CCA-mult method, the influence of the features on the components learned appears to be less compared to the features' influence when we use the CCA-concat method, and this suggests that because we have more datasets corresponding to the fMRI datasets (from one in the CCA-concat case to 9, 11, or 20, depending on the datasets being analyzed in the CCA-mult case), the influence of the fMRI datasets increases.

Let us now look at the groupings that exist in these two tables. The first component for the WP and the WP-WO datasets show the tool and shelter dimensions, while for the WO dataset the first component shows the word-length dimension (short words at one end and long words at the other end). We also see the animal dimensions in component 2 of the WP and the WP-WO datasets, while there are more tool dimensions in component 3 of the WP dataset. The transportation dimension is present in component 4 of the WP dataset, while component 4 of the WP-WO dataset shows the apparel dimension. Note that there are some overlap between some of these factors and the factors found in case study 1. In particular, we also see the tool, shelter, word-length, and apparel dimensions in case study 1.

		component 1	component 2	component 3	component 4	component 5
WP	positive	knife	bear	cat	truck	igloo
		screwdriver	dog	horse	car	cup
		spoon	cow	bee	train	corn
		pliers	beetle	fly	airplane	door
		hammer	ant	hand	foot	key
	negative	apartment	glass	saw	refrigerator	arm
		barn	chair	screwdriver	shirt	foot
		church	pants	spoon	bear	coat
		house	leg	knife	telephone	horse
		closet	bottle	pliers	dog	hand
WO	positive	refrigerator	pliers	pliers	corn	hand
		screwdriver	carrot	arm	barn	key
		telephone	screwdriver	hand	cup	spoon
		butterfly	lettuce	arch	cow	bear
		apartment	celery	hammer	igloo	train
	negative	eye	desk	dog	pants	skirt
		cat	car	cat	horse	table
		ant	house	cow	hand	pants
		leg	church	beetle	truck	shirt
		cup	bed	bee	bicycle	bottle
WP-WO	positive	knife	screwdriver	butterfly	dress	cup
		spoon	refrigerator	beetle	pants	igloo
		saw	pliers	telephone	coat	corn
		pliers	hammer	bear	horse	bell
		cat	bottle	lettuce	skirt	tomato
	negative	apartment	cat	leg	spoon	hand
		church	bee	chair	saw	foot
		closet	fly	key	car	horse
		house	car	door	bed	arm
		dresser	eye	pants	arm	bicycle

Table 5.3.3: Stimulus word rankings of the first five components learned by the CCA-mult method when we use the 485verb features.

		component 1	component 2	component 3	component 4	component 5
WP	positive	knife	glass	cat	truck	igloo
		screwdriver	chair	horse	car	cup
		spoon	bottle	bee	train	corn
		pliers	pants	hand	airplane	tomato
		hammer	door	fly	foot	key
	negative	apartment	bear	saw	refrigerator	arm
		barn	cow	spoon	shirt	foot
		church	dog	screwdriver	bear	coat
		house	beetle	pliers	telephone	hand
		closet	ant	knife	dog	horse
WO	positive	refrigerator	pliers	dog	hand	pants
		apartment	screwdriver	cat	bicycle	skirt
		telephone	lettuce	cow	horse	shirt
		screwdriver	carrot	horse	pants	coat
		butterfly	celery	beetle	arm	dress
	negative	eye	car	pliers	cup	bear
		cat	desk	arch	barn	hand
		ant	house	key	igloo	key
		leg	bed	arm	corn	chimney
		bee	church	door	arch	train
WP-WO	positive	apartment	cat	leg	spoon	hand
		church	bee	chair	saw	foot
		closet	cow	key	car	horse
		house	fly	pants	arm	arm
		dresser	dog	foot	knife	bicycle
	negative	saw	screwdriver	butterfly	dress	igloo
		spoon	pliers	telephone	pants	cup
		knife	refrigerator	beetle	shirt	bell
		pliers	bottle	lettuce	coat	corn
		cat	hammer	bear	skirt	tomato

Table 5.3.4: Stimulus word rankings of the first five components learned by the CCA-mult method when we use the intel218 features.

PCA Let us now consider the top stimulus words extracted by the PCA method, shown in tables 5.3.5 and 5.3.6. Similar to the results of the CCA-mult method, here comparing the two tables we also see that the extracted words are roughly stable across the two tables. Unlike in the CCA-mult case, however, in this case, it appears that we reduce the influence of the predefined semantic features when we combine them with the fMRI datasets.

In terms of the semantic dimensions present, we also find the tool and shelter dimensions for the first component of the WP and the WP-WO datasets, while the first component of the WO dataset exhibits the word-length dimension. The animal dimension is present in the second and third components of the WP dataset, and the third component of the WO and WP-WO datasets. We also see the tool dimension again in the third component of the WP dataset and the fourth component of the WP-WO dataset.

		component 1	component 2	component 3	component 4	component 5
WP	positive	knife	leg	cat	refrigerator	igloo
		screwdriver	chair	bee	lettuce	house
		spoon	glass	horse	butterfly	carrot
		carrot	bottle	beetle	tomato	cup
		saw	door	butterfly	cup	car
	negative	apartment	bear	saw	truck	coat
		barn	dog	spoon	arm	foot
		closet	cow	screwdriver	fly	hand
		church	ant	hammer	foot	dress
		house	beetle	knife	car	arm
WO	positive	refrigerator	pliers	dog	shirt	bed
		apartment	screwdriver	cow	foot	pants
		airplane	celery	fly	lettuce	bicycle
		dresser	carrot	cat	bee	coat
		bicycle	lettuce	beetle	arm	eye
	negative	eye	car	arch	tomato	glass
		cat	desk	key	door	key
		bee	house	arm	cup	cup
		leg	bed	hand	spoon	chimney
		ant	door	chair	chisel	fly
WP-WO	positive	apartment	eye	beetle	saw	foot
		closet	car	dog	spoon	hand
		church	cat	bear	arm	arm
		house	ant	cow	screwdriver	bicycle
		train	leg	butterfly	hammer	dresser
	negative	knife	screwdriver	leg	cat	cup
		saw	refrigerator	chair	horse	tomato
		spoon	pliers	door	dress	igloo
		pliers	celery	key	skirt	door
		carrot	butterfly	closet	coat	house

Table 5.3.5: Stimulus word rankings of the first five components learned by the PCA method when we use the 485verb features.

		component 1	component 2	component 3	component 4	component 5
WP	positive	knife	chair	saw	refrigerator	igloo
		screwdriver	glass	spoon	lettuce	house
		spoon	leg	screwdriver	butterfly	cup
		carrot	bottle	hammer	window	carrot
		saw	door	pliers	pliers	tomato
	negative	apartment	bear	cat	truck	coat
		barn	dog	horse	car	foot
		closet	cow	bee	igloo	arm
		church	beetle	hand	fly	hand
		house	ant	butterfly	train	dress
WO	positive	apartment	car	dog	foot	bed
		refrigerator	desk	beetle	shirt	bicycle
		airplane	house	cow	arm	pants
		dresser	bed	fly	chair	butterfly
		bicycle	truck	bear	dresser	coat
	negative	eye	screwdriver	key	tomato	glass
		bee	pliers	arch	cup	cup
		cat	celery	knife	door	fly
		ant	carrot	chair	spoon	church
		leg	lettuce	arm	skirt	chimney
WP-WO	positive	apartment	eye	beetle	saw	cup
		closet	cat	dog	spoon	igloo
		church	car	butterfly	screwdriver	tomato
		house	ant	bear	arm	door
		train	bee	telephone	hammer	house
	negative	saw	screwdriver	leg	cat	foot
		knife	refrigerator	chair	horse	hand
		spoon	pliers	key	dress	arm
		carrot	celery	door	skirt	bicycle
		pliers	hammer	closet	chisel	dresser

Table 5.3.6: Stimulus word rankings of the first five components learned by the PCA method when we use the intel218 features.

5.3.3 Feature loadings

In this case study, because the semantic features are also used to learn the components, we can also look into how each component ranks the semantic features from both the 485verb and intel218 sets. This is shown in several tables that follow. For instance, we see that corresponding to the tool dimension, from the 485verb set we have verbs like "cut", "tip", and "grip", and from the intel218 set we have questions regarding whether an object can be manipulated through holding.

From the following tables, one regularity we see is that in the case of the CCA-concat method, for the first component, when we compare the entries across the major rows (WP, WO, WP-WO), we see that the entries are highly similar. For instance, considering the 485verb features, we see the following eight features (out of 10 total) for the first component in all three cases (WP, WO, and WP-WO): cut, tip, ring, grip, travel, near, repair, walk. This indicates that for the first component learned using the CCA-concat method, there seems to be a lot of influence from the predefined semantic features compared to the influence from the fMRI data. To a lesser extent, this appears exist also in the case of the PCA method, but not in the case of the CCA-mult method. However, for subsequent components, there does not exist such a regularity for all three methods.

		component 1	component 2	component 3	component 4	component 5
WP	positive	cut	open	lift	fancy	build
		tip	repair	travel	wear	see
		ring	walk	bare	love	work
		grip	lift	work	clear	stretch
		milk	stand	stretch	stretch	wear
	negative	travel	travel	milk	cut	race
		near	love	open	grip	tip
		repair	drive	build	build	ride
		walk	ride	live	tip	ring
		dance	eat	paste	repair	crash
WO	positive	cut	love	eat	open	open
		tip	see	milk	milk	wear
		ring	play	love	build	stretch
		grip	press	live	pop	fancy
		milk	wear	travel	clear	shop
	negative	travel	tip	open	travel	build
		near	repair	lift	ride	press
		build	grip	stand	lift	ring
		walk	paste	break	shop	play
		repair	ring	pump	work	lock
WP-WO	positive	cut	open	wear	travel	ride
		tip	repair	fancy	tip	ring
		grip	stand	break	ride	paste
		ring	pump	stretch	lift	fancy
		remove	tip	lift	grip	race
	negative	travel	love	repair	open	build
		near	drive	cut	milk	see
		walk	rescue	build	live	cut
		dance	milk	tip	dance	press
		repair	eat	ring	pop	stand

Table 5.3.7: Top- and bottom-ranked verbs based on loading weights out of the verbs in the 485verb features learned by the CCA-concat method.

		component 1	component 2	component 3	component 4	component 5
WP	positive	CAN YOU HOLD IT IN ONE HAND?	DOES IT HAVE INTERNAL STRUCTURE?	IS IT USED IN SPORTS?	IS IT FRAGILE?	DOES IT HAVE FOUR LEGS?
		CAN YOU HOLD IT?	IS IT FAST?	WOULD YOU FIND IN THE BATHROOM?	IS TALLER THAN IT IS WIDE/LONG?	IS IT CONSCIOUS?
		CAN YOU PICK IT UP?	DOES IT HAVE SOME SORT OF NOSE?	DOES IT COME IN PAIRS?	DOES IT HAVE AT LEAST ONE HOLE?	CAN YOU BUY IT?
		DO YOU HOLD IT TO USE IT?	DOES IT HAVE AT LEAST ONE HOLE?	IS IT FLESH-COLORED?	IS IT USED DURING MEALS?	DOES IT HAVE LEGS?
		IS IT LIGHTWEIGHT?	DOES IT HAVE A FACE?	WOULD YOU FIND IT IN AN OFFICE?	IS IT HOLLOW?	DOES IT HAVE A BACKBONE?
	negative	IS IT BIGGER THAN A MICROWAVE OVEN?	IS IT USUALLY INSIDE?	DOES IT HAVE A HARD INSIDE?	CAN IT CAUSE YOU PAIN?	IS IT USED FOR TRANSPORTATION?
		IS IT BIGGER THAN A LOAF OF BREAD?	DO YOU HOLD IT TO USE IT?	DOES IT HAVE A HARD OUTER SHELL?	IS IT MADE OF METAL?	CAN IT BREAK?
		IS IT TALLER THAN A PERSON?	CAN YOU HOLD IT?	DOES IT GO IN YOUR MOUTH?	IS IT A TOOL?	DOES IT CONTAIN SOMETHING ELSE?
		DOES IT HAVE CORNERS?	WOULD YOU FIND IT IN A HOUSE?	DOES IT HAVE SEEDS?	DO YOU INTERACT WITH IT?	IS IT A VEHICLE?
		DOES IT OPEN?	CAN YOU PICK IT UP?	IS IT A VEGETABLE / PLANT?	DOES IT HAVE A HARD INSIDE?	IS IT USED IN SPORTS?
WO	positive	IS IT BIGGER THAN A MICROWAVE OVEN?	DOES IT HAVE AT LEAST ONE HOLE?	IS IT USED IN SPORTS?	IS IT USED DURING MEALS?	DO YOU WEAR IT?
		IS IT TALLER THAN A PERSON?	DOES IT CONTAIN LIQUID?	WOULD YOU FIND IT IN AN OFFICE?	CAN IT KEEP YOU DRY?	IS IT CLOTHING?
		IS IT BIGGER THAN A LOAF OF BREAD?	DOES IT HAVE INTERNAL STRUCTURE?	WOULD YOU FIND IN THE BATHROOM?	COULD YOU FIT INSIDE IT?	CAN YOU BUY IT?
		IS IT BIGGER THAN A BED?	IS IT FAST?	DO YOU SEE IT DAILY?	DOES IT COME FROM A PLANT?	DO YOU HOLD IT TO USE IT?
		IS IT BIGGER THAN A CAR?	DOES IT HAVE SOME SORT OF NOSE?	DO YOU USE IT DAILY?	IS IT CLOTHING?	CAN YOU HOLD IT?
	negative	CAN YOU HOLD IT IN ONE HAND?	DO YOU HOLD IT TO USE IT?	CAN YOU EAT IT?	IS IT A TOOL?	IS IT PART OF SOMETHING LARGER?
		CAN YOU HOLD IT?	DOES IT HAVE A HARD OUTER SHELL?	DOES IT HAVE ROOTS?	DO YOU INTERACT WITH IT?	DOES IT HAVE A HARD OUTER SHELL?
		CAN YOU PICK IT UP?	DOES IT HAVE A HARD INSIDE?	IS IT TASTY?	IS IT MADE OF METAL?	CAN IT BREAK?
		DO YOU HOLD IT TO USE IT?	IS IT MANUFACTURED?	DOES IT HAVE SEEDS?	DOES IT MAKE A SOUND?	WOULD YOU FIND IT IN A GARDEN?
		CAN IT BE EASILY MOVED?	IS IT MANMADE?	IS IT A VEGETABLE / PLANT?	CAN IT CAUSE YOU PAIN?	DOES IT COME IN PAIRS?
WP-WO	positive	CAN YOU HOLD IT IN ONE HAND?	DOES IT HAVE INTERNAL STRUCTURE?	WOULD YOU FIND IN THE BATHROOM?	IS IT A TOOL?	IS IT USED FOR TRANSPORTATION?
		CAN YOU HOLD IT?	DOES IT HAVE AT LEAST ONE HOLE?	IS IT USED IN SPORTS?	CAN IT CAUSE YOU PAIN?	CAN IT BREAK?
		CAN YOU PICK IT UP?	DOES IT HAVE SOME SORT OF NOSE?	WOULD YOU FIND IT IN AN OFFICE?	IS IT MADE OF METAL?	IS IT USED TO CARRY THINGS?
		DO YOU HOLD IT TO USE IT?	IS IT FAST?	DO YOU SEE IT DAILY?	DO YOU INTERACT WITH IT?	IS IT A BODY PART?
		CAN IT BE EASILY MOVED?	DOES IT CONTAIN LIQUID?	DOES IT COME IN PAIRS?	DOES IT HAVE A HARD INSIDE?	DOES IT CONTAIN SOMETHING ELSE?
	negative	IS IT BIGGER THAN A MICROWAVE OVEN?	DO YOU HOLD IT TO USE IT?	DOES IT HAVE A HARD OUTER SHELL?	IS IT FRAGILE?	CAN YOU BUY IT?
		IS IT TALLER THAN A PERSON?	IS IT USUALLY INSIDE?	CAN YOU EAT IT?	IS IT USED DURING MEALS?	IS IT CONSCIOUS?
		IS IT BIGGER THAN A LOAF OF BREAD?	DOES IT HAVE A HARD INSIDE?	DOES IT HAVE ROOTS?	IS TALLER THAN IT IS WIDE/LONG?	DOES IT HAVE A FACE?
		IS IT BIGGER THAN A BED?	CAN YOU HOLD IT?	IS IT A VEGETABLE / PLANT?	IS IT HOLLOW?	DOES IT HAVE A BACKBONE?
		DOES IT HAVE CORNERS?	IS IT MANUFACTURED?	DOES IT GO IN YOUR MOUTH?	COULD YOU FIT INSIDE IT?	DO YOU WEAR IT?

Table 5.3.8: Top- and bottom-ranked questions based on loading weights out of the questions in the intel218 features learned by the CCA-concat method.

		component 1	component 2	component 3	component 4	component 5
WP	positive	cut	eat	ride	travel	milk
		tip	travel	love	lift	build
		grip	grip	play	drive	open
		fancy	battle	learn	stop	pop
		remove	attack	race	pick	blind
	negative	near	fancy	grip	love	bare
		walk	open	cut	tie	race
		dance	lift	open	battle	hold
		close	clear	remove	phone	reach
		repair	pop	near	blow	clear
WO	positive	repair	cut	pump	milk	travel
		tip	tip	open	dance	press
		camp	ring	hold	build	hold
		bang	grip	lock	pop	ring
		ride	eat	grip	near	throw
	negative	press	near	rescue	race	see
		play	travel	live	ride	wear
		dry	walk	love	wear	fancy
		milk	open	milk	rescue	stretch
		run	build	eat	shop	clear
WP-WO	positive	cut	tip	eat	fancy	milk
		tip	cut	pin	clear	build
		fancy	repair	battle	wear	challenge
		press	pump	ride	ride	paste
		grip	grip	phone	tie	open
	negative	near	play	lift	grip	hold
		walk	drive	open	travel	race
		repair	build	fancy	say	ride
		travel	love	press	remove	bare
		open	dry	break	build	lift

Table 5.3.9: Top- and bottom-ranked verbs based on loading weights out of the verbs in the 485verb features learned by the CCA-mult method.

		component 1	component 2	component 3	component 4	component 5
WP	positive	CAN YOU HOLD IT IN ONE HAND?	WOULD YOU FIND IN THE BATHROOM?	DOES IT HAVE AT LEAST ONE HOLE?	IS IT USED FOR TRANSPORTATION?	DOES IT HAVE A HARD INSIDE?
		CAN YOU HOLD IT?	IS TALLER THAN IT IS WIDE/LONG?	IS IT FRAGILE?	IS IT USED TO CARRY THINGS?	DOES IT HAVE A HARD OUTER SHELL?
		DO YOU HOLD IT TO USE IT?	IS IT USUALLY INSIDE?	WOULD YOU FIND IN THE BATHROOM?	DOES IT CONTAIN SOMETHING ELSE?	IS IT SMALLER THAN A GOLFBALL?
		CAN IT BE EASILY MOVED?	WOULD YOU FIND IT IN A HOUSE?	CAN IT STRETCH?	CAN IT CAUSE YOU PAIN?	DO YOU HOLD IT TO USE IT?
		CAN YOU PICK IT UP?	WOULD YOU FIND IT IN A RESTAURANT?	DO YOU WEAR IT?	DOES IT HAVE WHEELS?	DOES IT HAVE AT LEAST ONE HOLE?
	negative	DOES IT OPEN?	DOES IT HAVE INTERNAL STRUCTURE?	DOES IT HAVE A HARD INSIDE?	CAN YOU BUY IT?	CAN IT BEND?
		IS IT TALLER THAN A PERSON?	IS IT FAST?	IS IT MADE OF METAL?	IS IT FUZZY?	IS IT USED IN SPORTS?
		IS IT BIGGER THAN A MICROWAVE OVEN?	DOES IT HAVE SOME SORT OF NOSE?	CAN IT CAUSE YOU PAIN?	IS IT YELLOW?	IS IT FLESH-COLORED?
		IS IT BIGGER THAN A LOAF OF BREAD?	DOES IT HAVE A FACE?	IS IT STRAIGHT?	DOES IT HAVE A BACKBONE?	IS IT PART OF SOMETHING LARGER?
		IS IT MADE OF WOOD?	DOES IT HAVE WIRES OR A CORD?	IS IT ALWAYS THE SAME COLOR(S)?	IS IT CONSCIOUS?	IS IT BIGGER THAN A LOAF OF BREAD?
WO	positive	DOES IT HAVE A FLAT / STRAIGHT TOP?	CAN YOU HOLD IT IN ONE HAND?	DOES IT MAKE SOUND CONTINUOUSLY WHEN ACTIVE?	IS IT USED IN SPORTS?	IS IT CLOTHING?
		IS IT STRAIGHT?	DO YOU HOLD IT TO USE IT?	DOES IT MAKE A SOUND?	CAN YOU SIT ON IT?	DO YOU WEAR IT?
		CAN IT CHANGE SHAPE?	CAN YOU HOLD IT?	DOES IT MAKE A NICE SOUND?	DO YOU INTERACT WITH IT?	CAN IT KEEP YOU DRY?
		WOULD YOU FIND IT IN THE SKY?	CAN YOU PICK IT UP?	DOES IT CONTAIN LIQUID?	CAN YOU RIDE ON/IN IT?	IS TALLER THAN IT IS WIDE/LONG?
		DOES IT OPEN?	IS IT LIGHTWEIGHT?	DOES IT HAVE INTERNAL STRUCTURE?	DOES IT COME IN PAIRS?	IS IT HOLLOW?
	negative	DOES IT LIVE ABOVE GROUND?	IS IT BIGGER THAN A LOAF OF BREAD?	WOULD YOU FIND IT IN AN OFFICE?	IS IT HOLLOW?	IS IT POINTED / SHARP?
		CAN IT CAUSE YOU PAIN?	DO YOU SEE IT DAILY?	DOES IT COME IN PAIRS?	WOULD YOU AVOID TOUCHING IT?	IS IT A TOOL?
		CAN YOU WALK ON IT?	IS IT MADE OF WOOD?	IS IT USUALLY INSIDE?	CAN YOU EAT IT?	DOES IT HAVE A HARD INSIDE?
		WOULD YOU FIND IN THE BATHROOM?	IS IT BIGGER THAN A MICROWAVE OVEN?	DOES IT HAVE A HARD INSIDE?	IS IT BIGGER THAN A CAR?	DOES IT HAVE MOVING PARTS?
		WOULD YOU FIND IT IN A RESTAURANT?	IS IT FURNITURE?	WOULD YOU FIND IN THE BATHROOM?	IS IT A BUILDING?	IS IT MADE OF METAL?
WP-WO	positive	IS IT BIGGER THAN A MICROWAVE OVEN?	IS IT FAST?	WOULD YOU FIND IN THE BATHROOM?	CAN IT CAUSE YOU PAIN?	IS IT USED IN SPORTS?
		IS IT BIGGER THAN A LOAF OF BREAD?	DOES IT HAVE AT LEAST ONE HOLE?	CAN YOU WALK ON IT?	DOES IT HAVE A HARD INSIDE?	CAN YOU SIT ON IT?
		IS IT TALLER THAN A PERSON?	DOES IT HAVE INTERNAL STRUCTURE?	WOULD YOU FIND IT IN A RESTAURANT?	IS IT A TOOL?	DO YOU INTERACT WITH IT?
		DOES IT OPEN?	DOES IT HAVE A FACE?	WOULD YOU FIND IT IN A ZOO?	IS IT MADE OF METAL?	IS IT PART OF SOMETHING LARGER?
		IS IT MADE OF WOOD?	CAN IT RUN?	WOULD YOU FIND IT IN AN OFFICE?	IS IT POINTED / SHARP?	IS IT USED FOR TRANSPORTATION?
	negative	CAN YOU HOLD IT IN ONE HAND?	IS IT STRAIGHT?	WOULD YOU FIND IT IN THE SKY?	DO YOU WEAR IT?	WOULD YOU AVOID TOUCHING IT?
		DO YOU HOLD IT TO USE IT?	IS IT USUALLY INSIDE?	DOES IT HAVE WIRES OR A CORD?	IS IT CLOTHING?	IS IT HOLLOW?
		CAN YOU HOLD IT?	IS IT A KITCHEN ITEM?	CAN YOU BUY IT?	IS IT FRAGILE?	DOES IT HAVE AT LEAST ONE HOLE?
		CAN YOU PICK IT UP?	IS TALLER THAN IT IS WIDE/LONG?	DOES IT LIVE IN GROUPS?	CAN IT STRETCH?	IS IT SLIPPERY?
		CAN IT BE EASILY MOVED?	IS IT FLAT?	DOES IT HAVE INTERNAL STRUCTURE?	IS TALLER THAN IT IS WIDE/LONG?	CAN YOU EAT IT?

Table 5.3.10: Top- and bottom-ranked questions based on loading weights out of the questions in the intel218 features learned by the CCA-mult method.

		component 1	component 2	component 3	component 4	component 5
WP	positive	cut	open	love	open	build
		tip	fancy	ride	ring	milk
		ring	lift	play	milk	live
		grip	break	rescue	paste	open
		milk	clear	race	eat	pop
	negative	travel	travel	open	travel	clear
		near	build	repair	drive	tip
		repair	eat	cut	lift	fancy
		walk	see	grip	ride	stand
		open	love	near	crash	bare
WO	positive	repair	cut	love	shop	ride
		travel	tip	milk	work	travel
		walk	ring	eat	lift	wear
		crash	grip	rescue	repair	race
		ride	eat	live	camp	stretch
	negative	press	travel	open	paste	build
		play	near	tip	open	milk
		milk	build	pump	tip	press
		love	open	lock	ring	challenge
		wear	see	lift	lock	walk
WP-WO	positive	travel	see	love	build	lift
		near	build	eat	grip	bare
		repair	press	travel	near	work
		walk	play	ride	remove	hold
		open	travel	battle	travel	stand
	negative	cut	tip	open	fancy	milk
		tip	cut	lift	wear	open
		ring	grip	break	clear	paste
		grip	repair	stand	love	build
		milk	ring	lock	tie	live

Table 5.3.11: Top- and bottom-ranked verbs based on loading weights out of the verbs in the 485verb features learned by the PCA method.

		component 1	component 2	component 3	component 4	component 5
WP	positive	CAN YOU HOLD IT IN ONE HAND?	IS IT MANMADE?	IS IT MADE OF METAL?	CAN YOU BUY IT?	DOES IT CONTAIN LIQUID?
		CAN YOU HOLD IT?	IS IT MANUFACTURED?	DOES IT HAVE A HARD INSIDE?	DO YOU HOLD IT TO USE IT?	DOES IT HAVE A HARD OUTER SHELL?
		CAN YOU PICK IT UP?	WAS IT INVENTED?	IS IT MANMADE?	CAN YOU HOLD IT IN ONE HAND?	IS PART OF IT MADE OF GLASS?
		CAN IT BE EASILY MOVED?	CAN YOU USE IT?	IS IT MANUFACTURED?	CAN YOU HOLD IT?	IS IT HOLLOW?
		DO YOU HOLD IT TO USE IT?	DO YOU USE IT DAILY?	DOES IT HAVE A HARD OUTER SHELL?	CAN YOU PICK IT UP?	CAN YOU EAT IT?
	negative	IS IT BIGGER THAN A MICROWAVE OVEN?	DOES IT GROW?	IS IT ALIVE?	IS IT USED FOR TRANSPORTATION?	IS IT USED IN SPORTS?
		IS IT TALLER THAN A PERSON?	IS IT ALIVE?	IS IT SOFT?	DOES IT CONTAIN SOMETHING ELSE?	CAN IT BEND?
		IS IT BIGGER THAN A LOAF OF BREAD?	WAS IT EVER ALIVE?	WAS IT EVER ALIVE?	IS IT USED TO CARRY THINGS?	DOES IT COME IN PAIRS?
		DOES IT OPEN?	DOES IT HAVE SOME SORT OF NOSE?	DOES IT GROW?	IS IT FAST?	DO YOU INTERACT WITH IT?
		IS IT BIGGER THAN A BED?	DOES IT CONTAIN LIQUID?	DOES IT HAVE AT LEAST ONE HOLE?	CAN YOU RIDE ON/IN IT?	IS IT FLESH-COLORED?
WO	positive	IS IT MANMADE?	IS IT BIGGER THAN A MICROWAVE OVEN?	DOES IT CONTAIN LIQUID?	IS IT USED IN SPORTS?	DO YOU LOVE IT?
		IS IT MANUFACTURED?	IS IT BIGGER THAN A LOAF OF BREAD?	DOES IT HAVE SOME SORT OF NOSE?	DOES IT COME IN PAIRS?	IS IT SOFT?
		IS IT BIGGER THAN A MICROWAVE OVEN?	IS IT TALLER THAN A PERSON?	DOES IT HAVE A FACE?	WOULD YOU FIND IT IN THE FOREST?	IS IT USED IN SPORTS?
		WAS IT INVENTED?	IS IT BIGGER THAN A BED?	IS IT AN ANIMAL?	DOES IT HAVE MOVING PARTS?	DOES IT HAVE LEGS?
		DOES IT HAVE A HARD OUTER SHELL?	CAN IT KEEP YOU DRY?	IS IT ALIVE?	CAN YOU SIT ON IT?	CAN YOU SIT ON IT?
	negative	IS IT ALIVE?	CAN YOU HOLD IT IN ONE HAND?	IS IT MANUFACTURED?	DO YOU HOLD IT TO USE IT?	DOES IT HAVE A HARD OUTER SHELL?
		WAS IT EVER ALIVE?	CAN YOU HOLD IT?	IS IT MANMADE?	CAN YOU BUY IT?	DOES IT HAVE A HARD INSIDE?
		CAN IT BEND?	CAN YOU PICK IT UP?	WOULD YOU FIND IT IN AN OFFICE?	DOES IT HAVE A HARD INSIDE?	IS IT BIGGER THAN A CAR?
		DOES IT GROW?	DO YOU HOLD IT TO USE IT?	DO YOU USE IT DAILY?	IS IT SILVER?	IS TALLER THAN IT IS WIDE/LONG?
		CAN YOU HOLD IT?	IS IT LIGHTWEIGHT?	DOES IT HAVE CORNERS?	IS IT MANMADE?	IS PART OF IT MADE OF GLASS?
WP-WO	positive	IS IT BIGGER THAN A MICROWAVE OVEN?	IS IT ALIVE?	DOES IT GROW?	DOES IT HAVE A HARD INSIDE?	IS IT HOLLOW?
		IS IT TALLER THAN A PERSON?	WAS IT EVER ALIVE?	IS IT ALIVE?	IS IT MADE OF METAL?	IS PART OF IT MADE OF GLASS?
		IS IT BIGGER THAN A LOAF OF BREAD?	DOES IT HAVE LEGS?	DOES IT HAVE SOME SORT OF NOSE?	CAN IT CAUSE YOU PAIN?	DOES IT HAVE A HARD OUTER SHELL?
		IS IT BIGGER THAN A BED?	CAN IT RUN?	WAS IT EVER ALIVE?	DOES IT HAVE A HARD OUTER SHELL?	IS IT BIGGER THAN A BED?
		DOES IT OPEN?	IS IT FAST?	DOES IT HAVE A FACE?	IS IT A TOOL?	DOES IT HAVE AT LEAST ONE HOLE?
	negative	CAN YOU HOLD IT IN ONE HAND?	IS IT MANUFACTURED?	IS IT MANMADE?	IS IT SOFT?	IS IT USED IN SPORTS?
		CAN YOU HOLD IT?	WAS IT INVENTED?	IS IT MANUFACTURED?	DO YOU WEAR IT?	DOES IT COME IN PAIRS?
		CAN YOU PICK IT UP?	IS IT MANMADE?	DOES IT HAVE CORNERS?	IS IT FRAGILE?	CAN YOU SIT ON IT?
		CAN IT BE EASILY MOVED?	DO YOU HOLD IT TO USE IT?	WAS IT INVENTED?	IS IT CLOTHING?	IS IT A BODY PART?
		DO YOU HOLD IT TO USE IT?	DOES IT HAVE A HARD OUTER SHELL?	DO YOU USE IT DAILY?	WOULD YOU FIND IN THE BATHROOM?	CAN IT BEND?

Table 5.3.12: Top- and bottom-ranked questions based on loading weights out of the questions in the intel218 features learned by the PCA method.

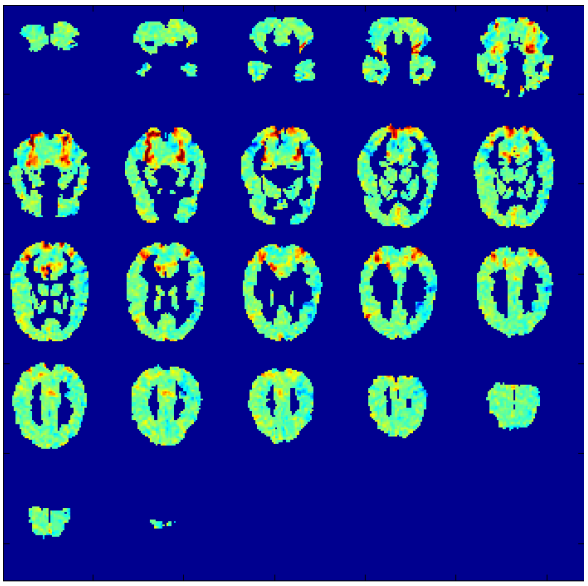
5.3.4 fMRI loadings

Now we look at how a particular component of each of our factor analytic methods projects to the whole brain. In particular, we focus on the first component. For each study, we pick a couple of subjects: subjects 1 and 5 for the WP study, and subjects 1 and 11 for the WO study; and we show the results of the methods when all the subjects from both studies are included. Note that subject 1 in both the WP and WO corresponds to the same person, and was referred to also as subject A in the previous chapter.

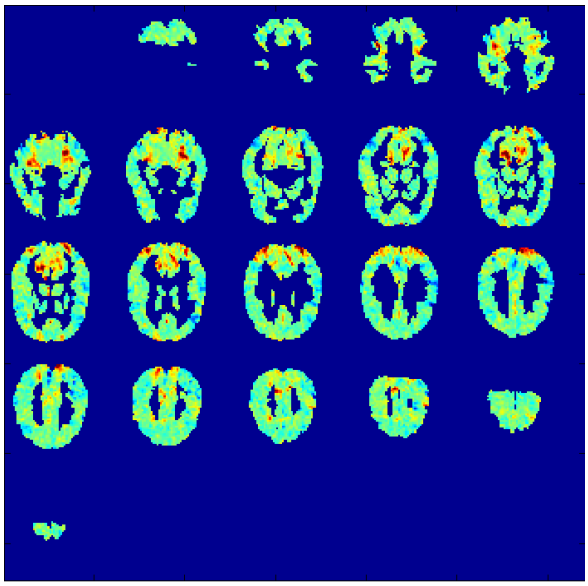
We first show a case where we jointly analyze both the WP and WO studies. Figure 5.3.2 shows the fMRI loadings for the first component for the four subjects when we use CCA-mult in conjunction with the intel218 features. We see that these loadings are highly similar to the ones for the first component in figures 4.2.8 and 4.2.9 from case study 1. These loadings are also highly similar to the loadings of the PCA method (figure 5.3.3), and to a lesser extent, to the loadings of the CCA-concat method (figure 5.3.4). Still to see the differences between the loadings of the CCA-concat method and the loadings of the other two methods requires a close scrutiny of the figures. This confirms what we see with the rankings of stimulus words and predefined semantic features, i.e. for the CCA-mult and PCA methods, in the WP-WO case, the contribution to the first component comes mainly from fMRI data. Furthermore, even for the CCA-concat method, for which we have found somewhat more influence from the predefined semantic features compared to the other two methods, we see that the loadings are also similar to those of the other two methods, indicating that the fMRI data still has some major contribution to the component, at least for the first component.

However, when we consider fMRI data only from a particular study, the method used has some influence on the fMRI loadings, especially in the case of WO dataset. This can be seen in figures 5.3.5, 5.3.6, and 5.3.7. Here we show the cases when we use the intel218 features. When we compare the loadings in these figures, we see notable differences among the loadings in the three cases (CCA-concat, CCA-mult, and PCA). Note that in the previous case study, we have found that when we analyze data from both WP and WO studies, the loadings of the first component are highly similar across methods, but when we analyze data from only the WO study, there are more differences in the loadings of the first component learned by each method. From what we have seen, that pattern appears to be also present in this case study.

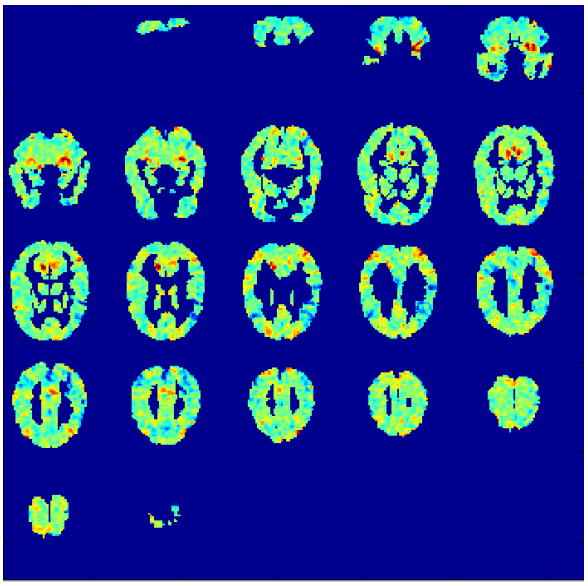
(a) subject 1WP



(b) subject 5WP



(c) subject 1WO



(d) subject 11WO

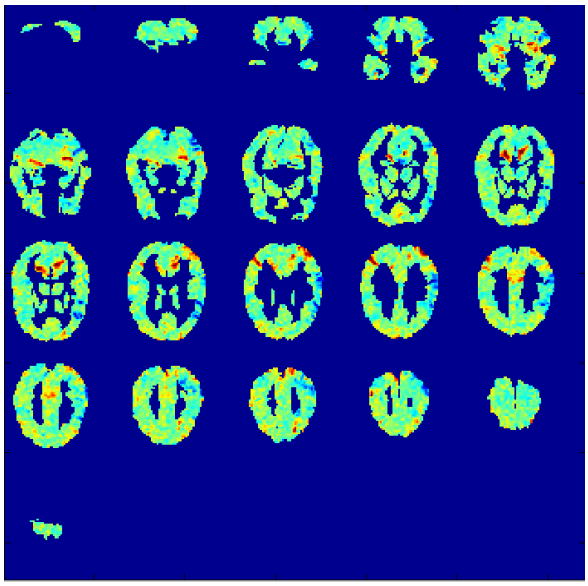
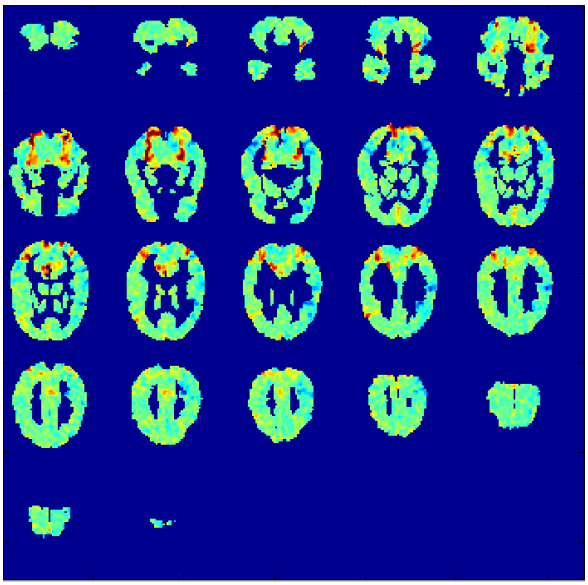
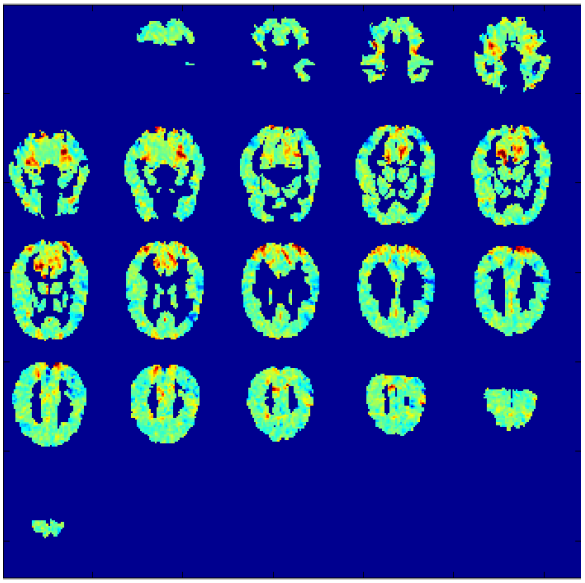


Figure 5.3.2: fMRI loadings of the first component for subjects 1WP, 5WP, 1WO, 11WO, learned by the CCA-mult-comb method in conjunction with the intel218 features.

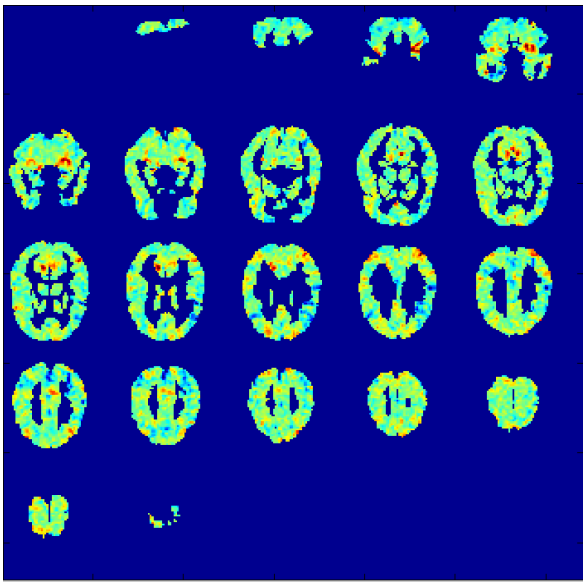
(a) subject 1WP



(b) subject 5WP



(c) subject 1WO



(d) subject 11WO

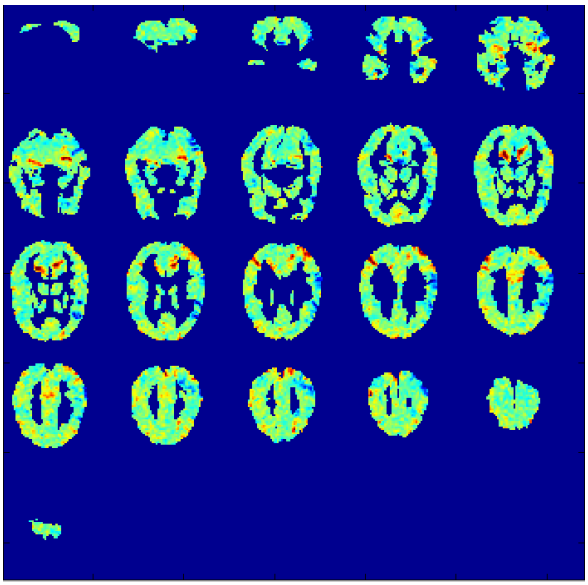
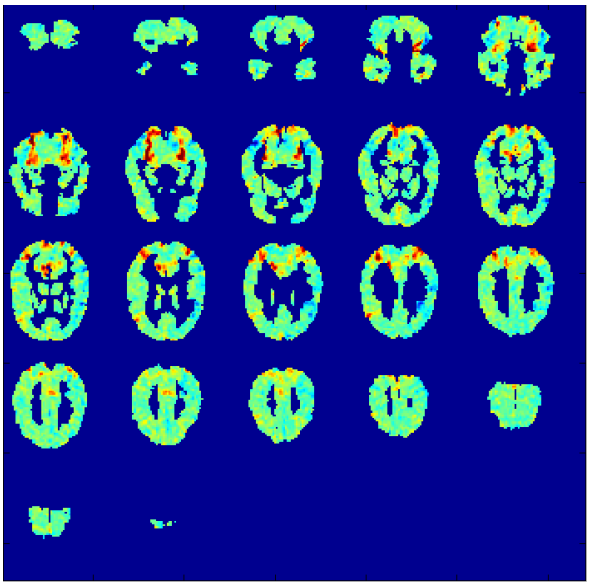
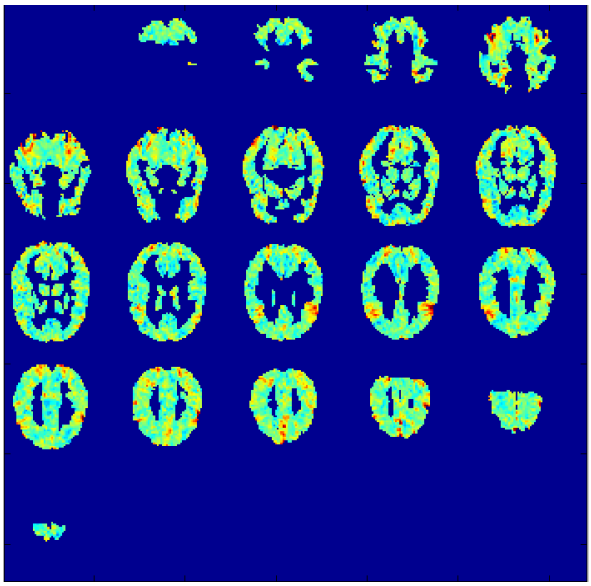


Figure 5.3.3: fMRI loadings of the first component for subjects 1WP, 5WP, 1WO, 11WO, learned by the PCA-comb method in conjunction with the intel218 features.

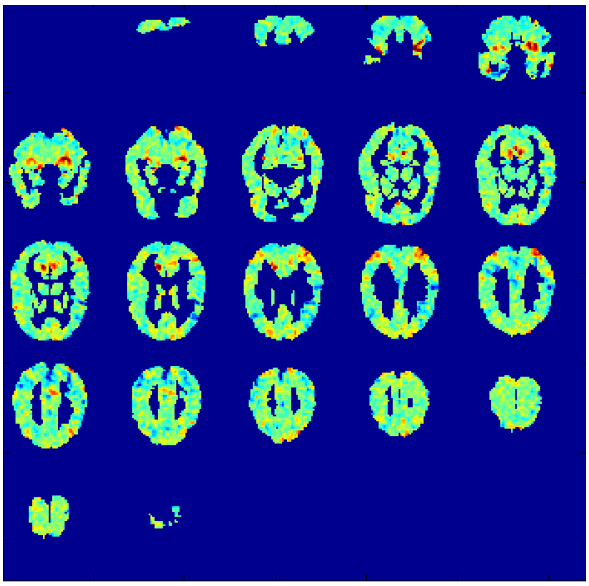
(a) subject 1WP



(b) subject 5WP



(c) subject 1WO



(d) subject 11WO

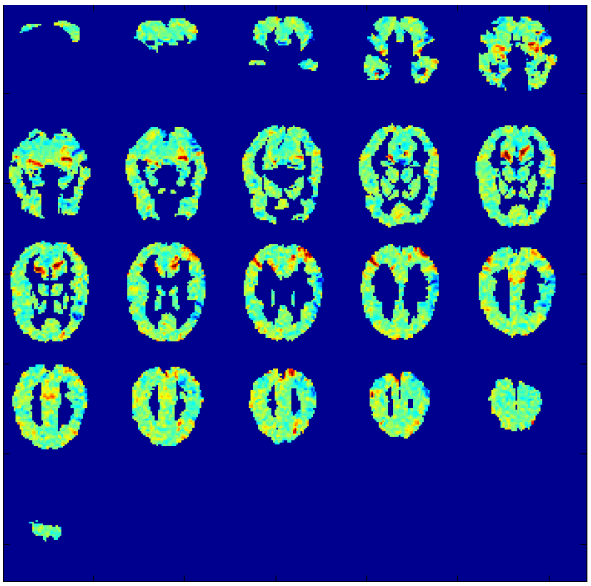


Figure 5.3.4: fMRI loadings of the first component for subjects 1WP, 5WP, 1WO, 11WO, learned by the CCA-concat-comb method in conjunction with the intel218 features.

(a) subject 1WO

(b) subject 11WO

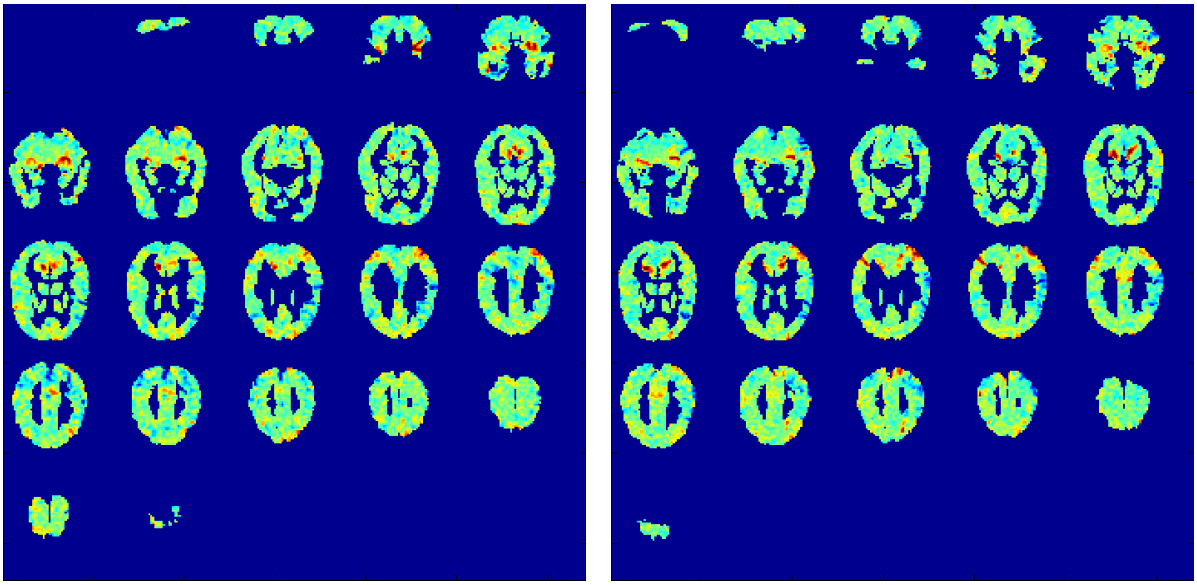


Figure 5.3.5: fMRI loadings of the first component for subjects 1WO, 11WO, learned by the CCA-concat-WO method in conjunction with the intel218 features.

(a) subject 1WO

(b) subject 11WO

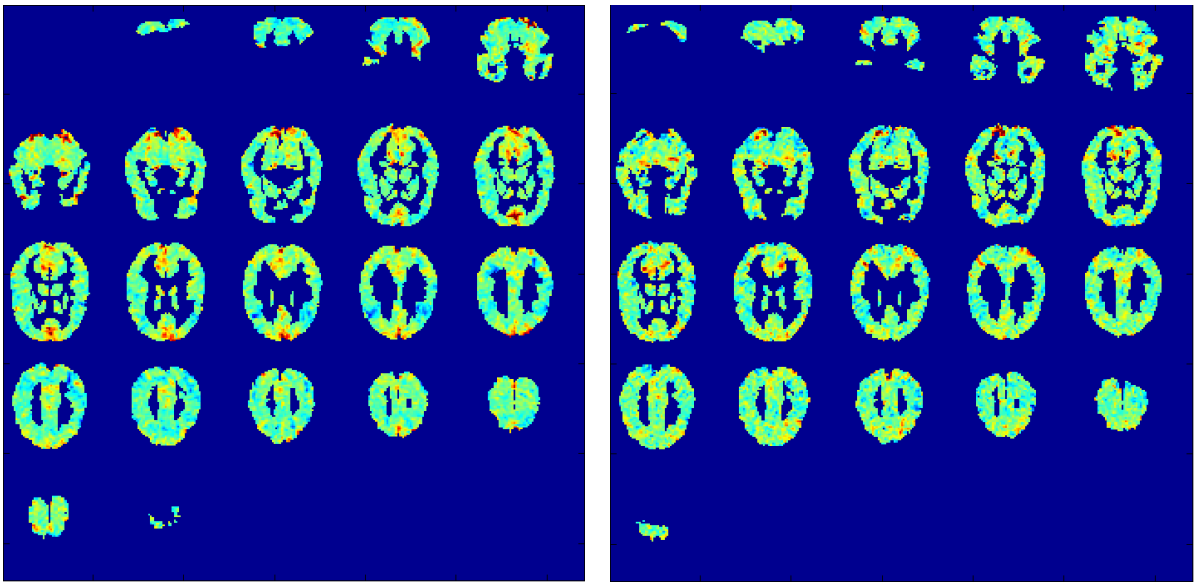


Figure 5.3.6: fMRI loadings of the first component for subjects 1WO, 11WO, learned by the CCA-mult-WO method in conjunction with the intel218 features.

(a) subject 1WO

(b) subject 11WO

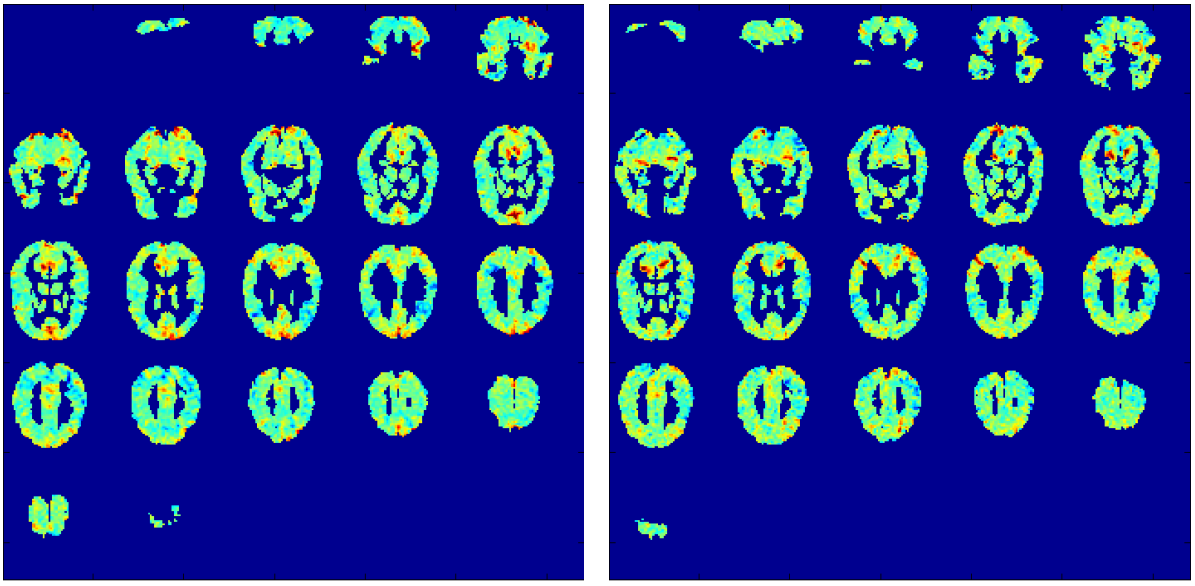


Figure 5.3.7: fMRI loadings of the first component for subjects 1WO, 11WO, learned by the PCA-WO method in conjunction with the intel218 features.

5.4 Discussion

We revisit the questions that this case study tries to address and discuss the answers shown by the experimental results:

1. Can we get better prediction, as measure by prediction accuracies, using the latent-factor model as opposed to the baseline model, for different kinds of predefined semantic features?

   The experimental results show that the answer is yes. In particular, when we use the CCA-mult method, in three cases we obtain significantly better predictive accuracies compared to the baseline model and in one case (intel218-WO), its accuracies match those of the baseline method. The other two methods considered, however, the CCA-concat and PCA methods, do not significantly outperform the baseline method.

2. How do the three methods used to estimate the parameters of the latent-factor model compare to one another?

   Based on the predictive accuracies, the best method is the CCA-mult method, followed by the CCA-concat method, and the PCA method has the worst predictive accuracies.

3. How do the prediction accuracies vary with the number of components?

   In a majority of the cases, the best prediction accuracies are obtained when five components are used.

4. How would the prediction accuracies be affected if we use fMRI data from multiple studies as opposed to fMRI data from a single study?

   In one case, the 485verb-WO case, we see improvements when fMRI data from multiple studies are used compared to when fMRI data from only one study are used. In the other cases, however, there are no significant difference between the accuracies when using multiple-study fMRI data vs when using single-study fMRI data. One reason for the cases with no significant differences between single-study and multiple-study analysis might be the fact that most of the relevant information for generalizing the fMRI activations for new words is captured in the combination of the predefined semantic features along with the particular study's fMRI data; only when this is not the case might we see improvements in predictive accuracies when we combine fMRI data across multiple studies.

5. How do the prediction accuracies compare with different kinds of predefined semantic features?

   The prediction accuracies obtained when using the intel218 features are better compared to the accuracies obtained when using the 485verb features. This is true across methods and across datasets.

6. What kind of semantic information, if any, is contained within each component? How does this information vary if we use data from multiple studies, and if we use different kinds of predefined semantic features?

   It is fairly consistent that when the WP dataset is involved, the first component extracted contains the shelter dimension, regardless of methods. On the other hand, when the WO dataset is analyzed on its own, we see the word-length dimension reflected in one of the first few components, with the exception of the CCA2way-intel218 case. Other groupings that exist in some cases include groupings of animal words, groupings of tool words, groupings of apparel words, and groupings of words for body parts.

   When the WP and WO datasets are jointly analyzed, in most cases, the first component contains the shelter dimension. On the other hand, the word-length dimension present in some cases when the WO dataset is analyzed on its own gets reduced or disappears altogether.

   For the CCA-concat method and for each kind of predefined semantic features (485verb or intel218), there are a lot of similarities in the top/bottom predefined semantic features for the first component across the WP, WO, and WP-WO study cases. This indicates that the influence for the first component comes mainly from the predefined semantic features. The influence of the predefined semantic features seems to also exist, albeit to a lesser extent, when we use the PCA method. There are more variations in the top/bottom predefined semantic features across the study cases for the first com-

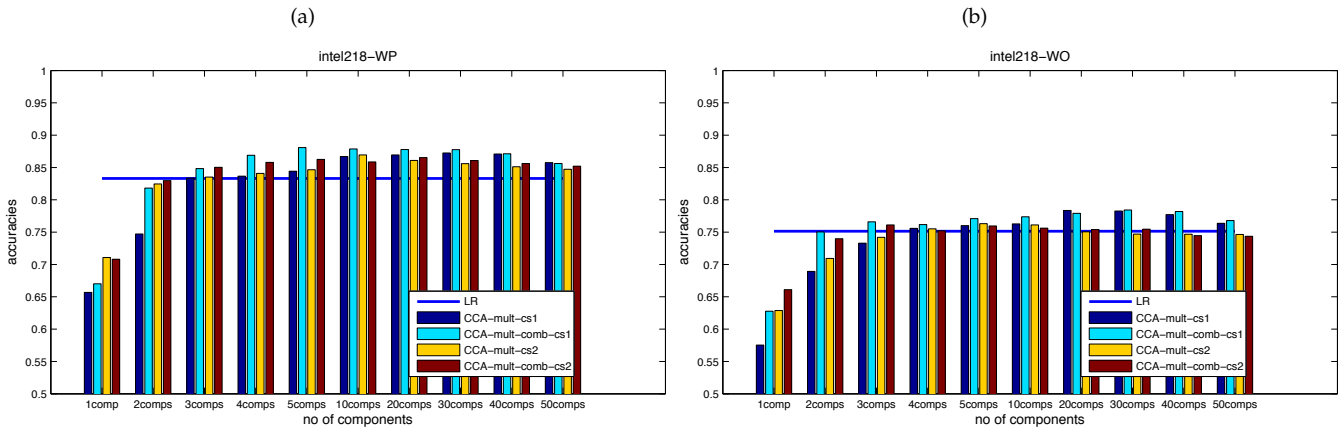


Figure 5.4.1: The accuracies of the CCA-mult methods in case study 1 and case study 2.

ponent when we use the CCA-mult method, indicating less influence from the predefined semantic features compared to the other methods. For subsequent components, there are more variations in the top/bottom predefined semantic features across the three study cases for all three methods.

7. How is the information contained in a component reflected in the brain for each subject?

   The loadings of the first component are highly similar to those we obtain in case study 1.

8. How do the accuracies compare with those in case study 1?

   For this question, we focus on the CCA-mult methods in both case study 1 and case study 2 when we use the intel218 features. Figure 5.4.1 shows the accuracies. Here we can see the accuracies of the methods in case study 2 are somewhat worse compared to the accuracies in case study 1. Also, the peak accuracies obtained in case study 2 occur when we use fewer components compared to the number of components yielding the peak accuracies in case study 1. There are two major differences between case study 1 and case study 2:

   - In case study 1, the model is learned in two stages, first learning the common factors across the fMRI data, and then in the second stage, we learn the linear regression coefficients from the semantic/base features to the learned common factors; in case study 2, the model is learned in one stage only: we learn the factors common across both fMRI data and the semantic features.
   - In case study 2, we normalize both the fMRI data and the semantic features so that each feature vector for a specific instance has length one. This is not performed in case study 1 when we use the intel218 features (in case study 1, we do normalize the 485verb features so that each instance has length one).

   The normalization step is motivated by the need to have similar scales in the datasets to analyze jointly, especially since in their raw forms the 485verb features have large magnitudes. However, we have also found that the normalization can change the accuracies significantly. To see this, in figure 5.4.2, we show the accuracies as figure 5.4.1, except in this, we do not normalize either the fMRI data or the semantic features.

   In this case, we see the accuracies in case study 2 get to be comparable to those in case study 1 while still reaching the peak with fewer number of components in case study 2, especially when we analyze the WP and WO datasets together. This result gives an indication of the need to explore the issue of normalization further when performing the scheme used in case study 2.

   Regardless of whether normalization is performed or not, another characteristic of the accuracies in case study 2 is that they exhibit steeper declines as the number of components increases. This is also

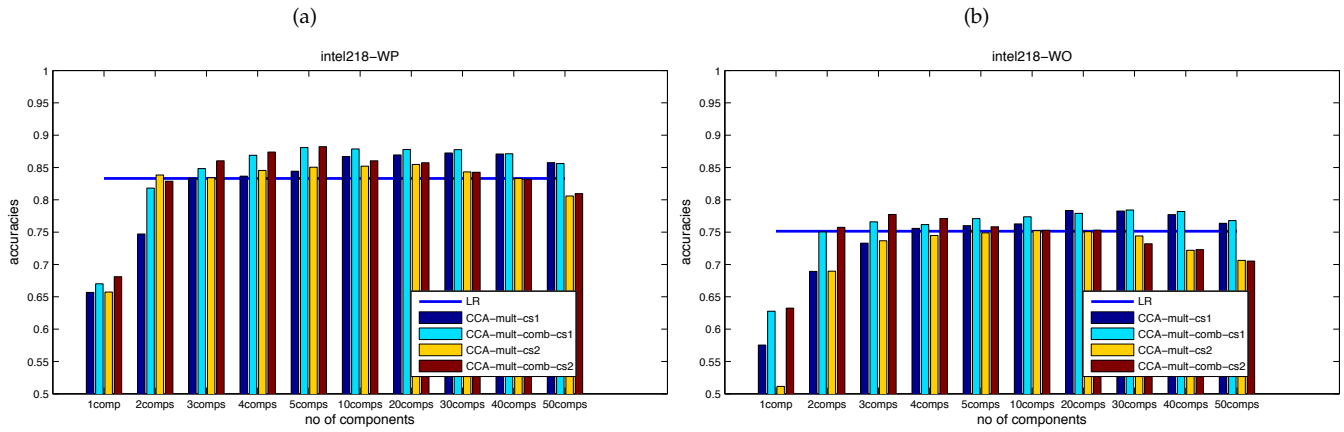


Figure 5.4.2: The accuracies of the CCA-mult methods in case study 1 and case study 2, not performing any normalization for case study 2.

seen when we use the 485verb features. It suggests a greater tendency to over-fit more when we learn common features using both fMRI data and the semantic/predefined semantic features.

5.5 Summary

In this chapter, we have seen another application of the linear factor analysis framework in which, instead of learning common factors across the fMRI datasets only, we learn common factors across the fMRI datasets and a dataset describing semantic features of the words/instances. Again, we see that in the predictive task considered here, this approach can yield significantly better accuracies compared to the accuracies of the baseline method. When we compare the results we obtain in this chapter to those we obtain in case study 1, we see the accuracies of the method considered in this chapter to be comparable or slightly worse compared to those from case study 1.

Note, however, that there are aspects that we have yet to explore. In particular, we do not explore ways to adjust the influences from both the fMRI datasets and the predefined semantic features and whether those ways can have significant effects on the accuracies obtained. One way to do this adjustment in the context of methods that incorporate CCA is to adjust the regularization parameter for each dataset. In both case studies, we have set these parameters to 0.5 and we do not explore varying the values of these parameters. It is certainly conceivable that this regularization parameter setting are not optimal and we can obtain better accuracies using a different setting. It is also conceivable that the accuracies of the methods that include the predefined semantic features become significantly better when we use the optimal parameter setting for each method in both case studies. However, we leave the investigation of this as a future work.

Chapter 6

Conclusion

Let us first revisit the main thesis of the work presented here, originally stated in chapter 1:

It is possible to invent machine learning and statistical techniques that can combine data from multiple subjects and studies to improve predictive performance, such that common patterns of activations can be distinguished from subject-specific and/or study-specific patterns of activations.

Let us now relate this thesis with the results that we have shown in this work:

- In chapter 2, using the hierarchical linear model, we extend the Gaussian Naïve Bayes (GNB) classifier so that the resulting classifier—called the hierarchical Gaussian Naïve Bayes (HGNB) classifier—can be trained using fMRI data from multiple subjects. We see that when the number of training instances for a specific subject is low, the HGNB is able to leverage data from the other subjects, and as the number of training instances for a specific subject increases, it increases the weight of the subject's contribution and its performance converged to that of the GNB classifier trained on the same subject's data. When dealing with fMRI data, we typically face the situation where the number of data points for each subject is relatively low (especially compared to the number of dimensions in the data). This is the situation where we might expect the available data might not be sufficient to lead to an optimal subject-specific GNB classifier, and where the HGNB classifier is especially useful because it allows us to combine fMRI data from multiple subjects to achieve better accuracies (compared to the accuracies of both the subject-specific and pooled GNB classifiers). Also with respect to the main thesis, this HGNB classifier allows us to obtain, for a particular location in the brain, the parameter associated with that location that is common across all the subjects (in the hierarchical linear model terminology, the fixed effects), and how this parameter varies across subjects (captured by the random effects in the context of the hierarchical linear model). A requirement of the HGNB classifier is that each subject's fMRI data are registered to a common brain.
- We also use the hierarchical linear model (in chapter 2 again), as a way to combine fMRI data from multiple subjects in the context of prediction formulated as a linear regression problem. Despite the fact that in this context, the experimental results do not show clear advantage over subject-specific and pooled analyses, we still consider the model to be a potentially effective way to combine fMRI data across multiple subjects in the context of prediction using linear regression. In particular, our experimental results indicate that the use of a more informative prior on some of the parameters, for instance, the parameter for the covariance matrix of each subject's voxel activations, can potentially lead to significant improvements in predictive accuracies. Like in the HGNB case, the model also has the requirement that each subject's fMRI data are registered to a common brain.
- The models based on the hierarchical linear model assume that the multiple fMRI datasets have the same kinds of dimensions, requiring these various fMRI datasets to be registered to a common brain. This requirement might be too restrictive for fMRI data given the anatomical and functional variabilities that exist in the brains of different individuals. To avoid this restriction, in chapter 3, we propose combining fMRI data from multiple subjects and studies using the linear factor analysis framework,

where we assume some underlying factors common across the multiple subjects and studies' data. In light of the main thesis, the factors in the context of the linear factor analysis framework provide a way to quantify the common patterns of brain activations across subjects and studies. We use principal components analysis (PCA) and canonical correlation analysis (CCA) to find these common factors.

- We show in chapter 4 that using the linear factor analysis framework, with the common factors learned using CCA, we obtain significant improvements in predictive accuracies in a task of predicting fMRI activations associated with concrete objects. We verify that the improvements are due to the integration of fMRI data across subjects and/or studies. Hence this proves the statement in the main thesis that we can improve predictive performance by combining data across subjects and studies using the linear factor analysis framework, although in the case of combining data across studies, we rely on the fact that the studies have the same kinds of instances (what we call matching instances in the next bullet).
- We also propose a method to deal with the case when for some instances in a particular dataset (corresponding to a particular subject in a particular study), we cannot find matching instances in another dataset (corresponding to another subject from a potentially different study), for instance when there is an instance corresponding to the object "cat" in the data for subject A, while subject B's data do not have any instance corresponding to "cat". The proposed method, described in chapter 3, is based on the idea of imputing the missing instances (in the example, above, imputing the "cat" instance in subject B's data). The results (in chapter 4) show the potential of this imputation method especially when for any two datasets, we cannot match all the instances in one dataset with the instances in the other dataset. Hence, the method provides a reasonable first step toward combining data from more disparate studies.
- In the context of predicting the fMRI activations for concrete objects, we also consider finding factors common across both the fMRI data and some predefined semantic features, described in chapter 5. The experimental results show that doing this we can with fewer factors/components obtain accuracies comparable to those obtained when we learn factors common across the fMRI data only.

We have seen how the various experimental results presented in previous chapters validate the main thesis. Note that combining data across subjects and/or studies when doing predictive fMRI analysis is especially beneficial because we are dealing with a lot of dimensions while having a limited number of instances. From the perspective of predictive accuracy, when we have a sufficient number of instances for a particular subject's fMRI data, we do not expect significant improvements by being able to consider data from the other subjects and/or studies. Nonetheless, we conjecture that approaches that integrate fMRI data across subjects and/or studies might still be beneficial when there are a lot of instances because they might still reveal the commonalities and the differences among the different subjects and/or studies' brain activations. In particular, in the context of the approaches based on the linear factor analysis framework, the commonalities are captured by the common factors while the differences are captured by the loadings of the common factors for different subjects and/or studies.

Note that the idea of common factors across subjects is also present in the work described in Just et al. (2010), which we also mention in chapter 4. We now briefly compare and contrast the general linear factor analysis framework presented in the thesis with the approach described in Just et al. (2010). The approach of Just et al. (2010) finds common factors through two stages of orthogonal factor analysis (we describe orthogonal factor analysis in chapter 3). In the first stage, the approach finds lobe-specific factors in each subject (a total of five lobes) and in the second stage, factor analysis is applied to the lobe-specific factors for all the lobes in all the subjects to find the common factors. The use of the lobes in the approach of Just et al. (2010) can be considered as an implicit spatial registration procedure. In contrast, the general linear factor analysis framework requires no spatial registration of the fMRI data, either explicit or implicit. In a sense, the approach of Just et al. (2010) is more constrained, assuming that voxels in similar locations should exhibit similar activations, while the general linear factor analysis framework is more flexible. The constraint might help in narrowing the space of possible solutions. However, the flexibility of the general linear factor analysis framework can be an advantage when the assumption incorporated in the constraint is violated, although more investigation is needed to determine whether this is indeed the case. Another

point of contrast is the fact that there are two stages in the approach of Just et al. (2010) while in the general linear factor analysis framework there is only one stage involved. When comparing the factors resulting from the two-stage approach with those resulting from a single-stage factor analysis, Just et al. (2010) find that the factors are similar but the factors from the two-stage approach are more interpretable. This brings up the question of whether it makes sense to add additional stages to the general linear factor analysis framework, and if that is the case, how to do so effectively. At this point, the answer to this question is not clear and will be explored as part of future work.

Before this thesis, approaches for combining fMRI data across multiple subjects and/or studies rely on matching activations that occur at the same location in different subjects and/or studies. We have shown in this thesis, especially with the linear factor analysis framework, that this is not necessary and that relating the different subjects and studies' fMRI data using the common factors can yield significant improvements in predictive accuracies. In addition, these common factors can also give us additional insights into the commonalities and differences that are present across the different subjects and studies. We can see examples of this in chapter 4, where we can see how the same factor is expressed differently for different subjects in either the same study or different studies, and even for the same subject in different studies. In general, we believe that the idea of common factors can be beneficial when applied to other fMRI data analysis settings involving multiple subjects and/or studies.

Based on the discussion in the previous paragraph, the results of this thesis should be of interest to those who do predictive modeling of fMRI data. We also believe that the techniques described in this thesis are also applicable to brain imaging data obtained using other modalities, such as EEG and MEG, and should be of interest also to researchers working with these kinds of data. As mentioned in chapter 1, the topic of this thesis is closely related to the area of multitask learning or transfer learning, and from that perspective, the results of the thesis should be of interest as well since, especially in the context of the linear factor analysis framework, we have presented a technique for performing multitask learning when the feature spaces for the different tasks are different.

6.1 Design Space for the Factor-Based Approaches

In this thesis we have shown that in the context of the linear factor analysis framework applied to fMRI data, the canonical correlation analysis (CCA) is especially effective when used as a method to learn the factors. However, we do not claim that CCA is the optimal method, from a predictive point of view, to learn the common factors from fMRI data.In chapter 3, we present our rationales for narrowing our focus on CCA and the principal component analysis (PCA). Another consideration for us is the amount of computation necessary to estimate a particular method. These rationales and consideration do not necessarily mean that these methods always yield better predictive accuracies compared to other methods that can be used to learn the factors, including the two other methods considered in that chapter: the orthogonal factor analysis and the higher-order generalized singular value decomposition; in fact, Just et al. (2010) show that the orthogonal factor analysis can be an effective way to learn factors from some fMRI data. Nonetheless, it might be impractical to consider exhaustively each of the possible ways to learn the factors. Here we briefly discuss methods other than those described in chapter 3. Note that everything that follows immediately below can also be considered as a potential area to explore as part of future work.

A couple of Bayesian methods that can potentially be effective when used to learn the common factors are the Indian Buffet Process (Griffiths and Ghahramani (2006)) and the sparse Bayesian factor regression (West (2003)) models. Both models allow sparse factors and the automatic discovery of the optimal number of factors. However, originally proposed, both methods require the use of Markov Chain Monte Carlo methods for inference, which is computationally relatively expensive. The amount of computation necessary to estimate the method is an important consideration for us, especially given the amount of data we have and the intensive cross-validation involved. In fact, we did a few experiments using the Indian Buffet Process model, but we decided to focus elsewhere because of the computation involved. It might be the case that there is a way to speed up the computation of these methods using, for instance, variational methods (Jordan et al. (1999)).

Another method that shares the form given in equation (3.1) is the independent component analysis (ICA, Hyvärinen et al. (2001)). ICA is based on the idea of finding factors that are not necessarily Gaussian. It has been applied in several contexts to fMRI data (for instance, McKeown et al. (1998) and Beckmann and Smith (2004)), and an extension has been proposed to analyze multiple-subject fMRI data (Beckmann and Smith (2005)). However, this extension is based on the assumption that the multiple-subject fMRI data have been registered, i.e. the different subjects' fMRI data having the same feature space, and it might be interesting to consider an extension that can be applied to fMRI data that are not necessarily registered. It might also be interesting to still consider the components learned by the standard ICA (with one group only) when we concatenate the different datasets' data matrices into one data matrix.

Yet another method that can potentially be effective is a variant of the (2-way) canonical correlation analysis that replaces the use of correlation with mutual information (Yin (2004)). One advantage that this method might have over CCA is that it can potentially account for nonlinear dependencies among the different datasets. However, it is also computationally expensive, since it involves estimating the mutual information in a nonparametric fashion, and an extension to more than two datasets still needs to be investigated.

Besides trying other methods, one can also explore whether using the methods we explore in this thesis (PCA and CCA) but changing the ways we preprocess and/or transform the data might lead to better predictive performance. For instance, in this thesis, all the methods incorporating PCA have applied PCA to the raw data. We might obtain different results when preprocessing the data so that each variable has unit variance before applying PCA to them[1].

6.2 Future Work

There are several possible directions for future work. In this thesis, we have presented results of combining data from two fMRI studies that are highly similar using the linear factor analysis framework. In addition, we also consider an imputation method to deal with the case when we are combining data from fMRI studies with some non-matching instances. Nonetheless, the imputation method still requires that there are instances shared among the datasets that we are jointly analyzing. And furthermore, there seems to be some room for improvement for the imputation method. One possible direction for future work is to consider methods that can integrate more disparate datasets, in which there might be no overlapping instances at all.

Two particular approaches are discussed in appendix A. There we frame the model used in chapter 4 as a two-stage linear regression problem and in one case, formulate an objective function for this problem, leading to the regularized bilinear regression model, and in the other case, formulate a probabilistic model for the problem, leading to the conditional factor analysis model. An advantage of both these formulations is that the different fMRI datasets do not need to have matching instances and we do not need to resort to imputing the data for the missing instances in the case of the regularized bilinear regression model, or in the case of the conditional factor analysis model, the imputation step can be integrated into the estimation procedure within the EM algorithm. However, as can be seen in appendix A, the results are worse compared to the results we obtain in chapter 4, so some modifications or improvements to this formulation still need to be investigated.

Another idea to integrate disparate datasets is by extending the canonical correlation analysis (CCA) method described in chapter 3. In particular, we need an entity that can tie in the disparate datasets, where by entity we mean a source of data (can be other than fMRI data) that contains information about all the instances present. Continuing the paradigm of predicting the fMRI activations for unseen words used in some of the examples, we consider the problem of integrating the analysis of fMRI datasets associated with different kinds of words, for instance, jointly analyzing fMRI studies of concrete nouns used in this thesis with an fMRI study of abstract nouns, with no overlaps between the concrete and the abstract nouns. If the semantic features used cover all the concrete and abstract nouns, then these features provide a way to tie the fMRI datasets together, and therefore the set of semantic features can be used as the entity to tie the

[1]This example was brought up by Zoubin Ghahramani.

disparate datasets together. In this particular case, we can (in the future) apply CCA to the fMRI datasets and the semantic features as follows. The CCA solution is provided by the generalized eigenvalue problem shown in equation (3.44). This generalized eigenvalue problem involves matrices consisting of covariance terms. The cross-covariance terms of datasets with no overlapping instances can be set to the zero matrix. For instance, in our hypothetical case, there will be non-zero cross-covariance terms of each of the fMRI datasets with the semantic features, but there will be zero cross-covariance terms between a concrete-noun fMRI dataset and an abstract-noun fMRI dataset. We can then solve the resulting generalized eigenvalue problem and used its solution like what is described in the case studies.

We also mention a direction of future work at the end of chapter 5. In chapters 4 and 5, for those methods that can be regularized, we choose an arbitrary value of 0.5 as the regularization parameter. Although using this method we indeed obtain accuracies better than those of a baseline model that considers only subject-specific data, there is a potential for improvements that can be gained if we explore the space of the regularization parameters in order to find settings that give the optimal accuracies. In appendix B, we present the results of a preliminary investigation of the effect of various values of the regularization parameter of the model used in chapter 4.

Another direction for future work is to see whether we can integrate the hierarchical modeling used in the approaches based on the hierarchical linear model with the linear factor analysis framework. This is useful for instance when we want to say that the factors in different studies can be similar but they do not have to be exactly the same. We wonder whether this kind of approach can yield additional improvements in predictive accuracies. This might be straightforward to do if we frame the linear factor analysis framework as a probabilistic model, then we can generalize this framework using hierarchical Bayes techniques. However, although it might straightforward conceptually, one challenge when going to the probabilistic approach is computation. It is expected that a substantial effort needs to be expended in order to make sure that the model estimation is still reasonably fast when we work in the probabilistic setting. We also note that the approach described in Just et al. (2010) can also be considered as a kind of hierarchical factor analysis approach, although not necessarily in the probabilistic setting, and another direction is to investigate how to incorporate the idea presented in Just et al. (2010) with the approaches presented in this thesis.

From the hierarchical linear modeling to the linear factor analysis framework, we have gone from assuming that all the datasets have the same kinds of features to being indifferent about the features present in each dataset. Having a model that can get us in between these two extremes might also have the potential to lead to models with better predictive ability.

Appendix A

The Regularized Bilinear Regression and the Conditional Factor Analysis Models

The model shown in figure 4.1.2 consists of two linear transformations: the first transformation mapping the base features to the learned common features, and the second transformation mapping the learned common features to the brain activations data. In chapter 4, we mention that this model is learned in two stages:

1. Stage 1: learn the common features and the second transformation using CCA
2. Stage 2: learn the first transformation using linear regression

We consider the question of whether the model can be learned in one stage only. To approach this question, let us consider the model more formally.

The model in figure 4.1.2 can be formulated mathematically as follows:

$$\mathbf{g}_w = \mathbf{A}\mathbf{f}_w \tag{A.1}$$
$$\mathbf{y}_w^{(m)} = \mathbf{B}^{(m)}\mathbf{g}_w, \tag{A.2}$$

where $\mathbf{f}_w$ denotes the $I \times 1$ vector of base features for word w, $\mathbf{g}_w$ denotes the $J \times 1$ vector of common features for word w, $\mathbf{y}_w^{(m)}$ denotes the $D^{(m)} \times 1$ vector of brain activations for group m (for instance, a particular subject from a particular study), $\mathbf{A}$ denotes the $I \times J$ matrix representing the first transformation, and $\mathbf{B}^{(m)}$ denotes the $J \times D^{(m)}$ matrix representing the second transformation for group m. Note that $\mathbf{f}_w$ and $\mathbf{Y}_w^{(m)}, \forall m$ are given and the rest need to be estimated.

The two stages described above provide a way to estimate the unknown parameters. We now consider a couple of ways to combine the two stages into one: the regularized bilinear regression model and the conditional factor analysis model.

A.1 The Regularized Bilinear Regression Model

A way to estimate the unknown parameters in equations (A.1) and (A.2) is to formulate an objective and find the parameter values that optimizes this objective. Let us consider the sum of squared errors to be our objective. In particular, in our case we can try to minimize the sum of squared errors of $\mathbf{y}_w^{(m)}$ and $\mathbf{B}^{(m)}\mathbf{g}_w$, or formally, after substituting the right-hand side of equation (A.1) for $\mathbf{g}_w$,

$$\sum_{w=1}^{W}\sum_{m=1}^{M}(\mathbf{y}_w^{(m)} - \mathbf{B}^{(m)}\mathbf{A}\mathbf{f}_w)^T(\mathbf{y}_w^{(m)} - \mathbf{B}^{(m)}\mathbf{A}\mathbf{f}_w). \tag{A.3}$$

We can then try to find the parameters $\mathbf{A}$ and $\mathbf{B}^{(m)}, \forall m$ (we can find $\mathbf{g}_w$ once we find $\mathbf{A}$ by using equation (A.1)) that minimize this objective. However, we might also want to regularization terms to avoid over-fitting. In particular, using the standard L2 regularization, we obtain the optimization problem:

$$\min_{\mathbf{A},\{\mathbf{B}^{(m)}\}_{m=1}^{M}} \sum_{w=1}^{W}\sum_{m=1}^{M}(\mathbf{y}_w^{(m)} - \mathbf{B}^{(m)}\mathbf{A}\mathbf{f}_w)^T(\mathbf{y}_w^{(m)} - \mathbf{B}^{(m)}\mathbf{A}\mathbf{f}_w) + \lambda_\alpha||\mathbf{A}||_F^2 + \sum_{m=1}^{M}\lambda_\beta^{(m)}||\mathbf{B}^{(m)}||_F^2, \quad \text{(A.4)}$$

where $||\cdot||_F$ denotes the Frobenius norm of a matrix. λ_α and $\lambda_\beta^{(m)}$ are the regularization parameters for $\mathbf{A}$ and $\mathbf{B}^{(m)}$, respectively. We call this formulation the *regularized bilinear regression* formulation.

The solution to this optimization problem can be obtained using gradient-based optimization techniques. To use any of these techniques, we need to find what the gradients are. To obtain the gradients, first note that the objective in the optimization problem (A.4) can be rewritten as

$$\sum_{m=1}^{M}\left(\operatorname{tr}\mathbf{Y}^{(m)}(\mathbf{Y}^{(m)})^T - 2\operatorname{tr}(\mathbf{B}^{(m)})^T\mathbf{Y}^{(m)}\mathbf{F}^T\mathbf{A}^T + \operatorname{tr}\mathbf{A}\mathbf{F}\mathbf{F}^T\mathbf{A}^T(\mathbf{B}^{(m)})^T\mathbf{B}^{(m)}\right) + \lambda_\alpha \operatorname{tr}\mathbf{A}^T\mathbf{A} + \sum_{m=1}^{M}\lambda_\beta^{(m)}\operatorname{tr}(\mathbf{B}^{(m)})^T\mathbf{B}^{(m)}, \quad \text{(A.5)}$$

where

$$\mathbf{Y}^{(m)} = \left[\mathbf{y}_1^{(m)}\cdots\mathbf{y}_W^{(m)}\right] \quad \text{(A.6)}$$

$$\mathbf{F} = [\mathbf{f}_1\cdots\mathbf{f}_W]. \quad \text{(A.7)}$$

The derivatives of the regularization terms can be obtained by noting that for a matrix $\mathbf{X}$, $||\mathbf{X}||_F^2 = \operatorname{tr}\mathbf{X}^T\mathbf{X}$, and

$$\frac{\partial}{\partial\mathbf{X}}\operatorname{tr}\mathbf{X}^T\mathbf{X} = 2\mathbf{X}. \quad \text{(A.8)}$$

We can also obtain the following

$$\frac{\partial}{\partial\mathbf{A}}\operatorname{tr}(\mathbf{B}^{(m)})^T\mathbf{Y}^{(m)}\mathbf{F}^T\mathbf{A}^T = (\mathbf{B}^{(m)})^T\mathbf{Y}^{(m)}\mathbf{F}^T \quad \text{(A.9)}$$

$$\frac{\partial}{\partial\mathbf{B}^{(m)}}\operatorname{tr}(\mathbf{B}^{(m)})^T\mathbf{Y}^{(m)}\mathbf{F}^T\mathbf{A}^T = \mathbf{Y}^{(m)}\mathbf{F}^T\mathbf{A}^T \quad \text{(A.10)}$$

and

$$\frac{\partial}{\partial\mathbf{A}}\operatorname{tr}\mathbf{A}\mathbf{F}\mathbf{F}^T\mathbf{A}^T(\mathbf{B}^{(m)})^T\mathbf{B}^{(m)} = 2(\mathbf{B}^{(m)})^T\mathbf{B}^{(m)}\mathbf{A}\mathbf{F}\mathbf{F}^T \quad \text{(A.11)}$$

$$\frac{\partial}{\partial\mathbf{B}^{(m)}}\operatorname{tr}(\mathbf{B}^{(m)})^T\mathbf{A}\mathbf{F}\mathbf{F}^T\mathbf{A}^T(\mathbf{B}^{(m)})^T\mathbf{B}^{(m)} = 2\mathbf{B}^{(m)}\mathbf{A}\mathbf{F}\mathbf{F}^T\mathbf{A}^T. \quad \text{(A.12)}$$

We now have all the terms needed to obtain the gradients.

In the regularized bilinear regression, imputation is not necessary when there non-matching instances. To see why, let us assume that word w is missing for group m. What this means is that we can skip the sum of squared errors corresponding to word w and group m in equation (A.4). The gradients can also be adjusted accordingly, and we still can obtain estimates for the parameters $\mathbf{A}$ and $\mathbf{B}^{(m)}$.

There are a couple of major differences between how the common features are learned in the regularized bilinear regression formulation and how they are learned using CCA:

1. CCA learns common features one by one, and there is an order of importance of the common features, the first one being the most important, and so on. In the regularized bilinear regression, all the common features are learned simultaneously, and there is no order of importance of the common features.
2. CCA actually learns separate features for each group m, the average of which we take to be the common features. The regularized bilinear regression learns the same common features for all the groups.

Also note that when there is only one group ($M = 1$), principal components regression and partial least squares regression also provide a solution to the general formulation given by equations (A.1) and (A.2), but there do not exist formulations of these methods when $M > 1$. More details on these methods can be found in, for instance, section 3.5 of Hastie et al. (2009).

Next, we see how the regularized bilinear regression performs when used in some of the experiments that we describe earlier in this chapter.

A.1.1 Experimental results

We perform experiments using the regularized bilinear regression. The setup of the experiments is similar to the setup described in section 4.2. Here we use only the **intel218** semantic features and we consider only when we combine data across studies, i.e. we do not consider analysis of only within-study data. We also run experiments when there are some non-matching instances, leaving words in the cat-short sets (table 4.3.4).

We consider different versions of the regularized bilinear regression method based on the following possible parameter settings:

- number of features learned: 5, 10, 20 features
- λ_α: 0.1, 1, 10
- $\lambda_\beta^{(m)}$: 0.1, 1, 10

Here we have the same regularization parameter $\lambda_\beta^{(m)}$ for all groups, and we do not consider varying this parameter for different groups. The solution for the regularized bilinear regression is obtained using conjugate gradient[1].

Let us first consider the accuracies obtained when all the 60 words are used, shown in figure A.1.1. Here we see that in all configurations of the regularization parameters, the accuracies never reach significantly above 0.6, regardless of the number of learned common features used. Based on these results, the value of the regularized bilinear regression formulation is questionable, although we note that there are other configurations that we have not considered (for instance, other values besides those listed above for the regularization parameters) due to the computational demand for estimating the parameters. The accuracies when $\lambda_\beta = 10$ are relatively higher compared to the accuracies in the other cases. Also, based on figures A.1.1, we see that when $\lambda_\beta = 10$, the accuracies are relatively higher compared to the other cases. The effect of λ_α and the number of components, on the other hand, seems small. This brings up the question of what we will see if we consider larger values of λ_β.

Figures A.1.2 (for cat-short-1) and A.1.3 (for cat-short-2) show the accuracies when some words are left out. Again, in all cases, the accuracies do not get significantly above 0.6. The bumps in accuracies when $\lambda_\beta = 10$ are still somewhat visible, although the effect seems to be less compared to when all the 60 words are used.

Given the fact that the regularized bilinear regression is estimated in one stage only, compared to the models we consider earlier in the chapter that require two stages for estimation, one might expect the accuracies of the regularized bilinear regression to be better compared to the accuracies of the models we consider earlier. The results, however, do not validate this expectation. Why is this the case? One answer might be the fact that the constraints of the regularized bilinear regression formulation is different from the

[1]We use the conjugate gradient implementation provided by Carl Edward Rasmussen, available at http://www.kyb.tuebingen.mpg.de/bs/people/carl/code/minimize/. 200 iterations of conjugate gradient search are performed.

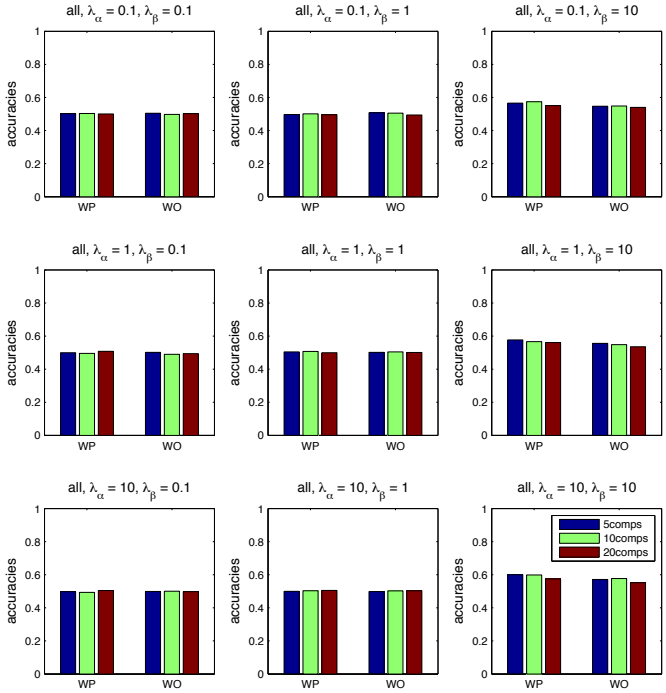


Figure A.1.1: The accuracies of the regularized bilinear regression when we use all 60 words.

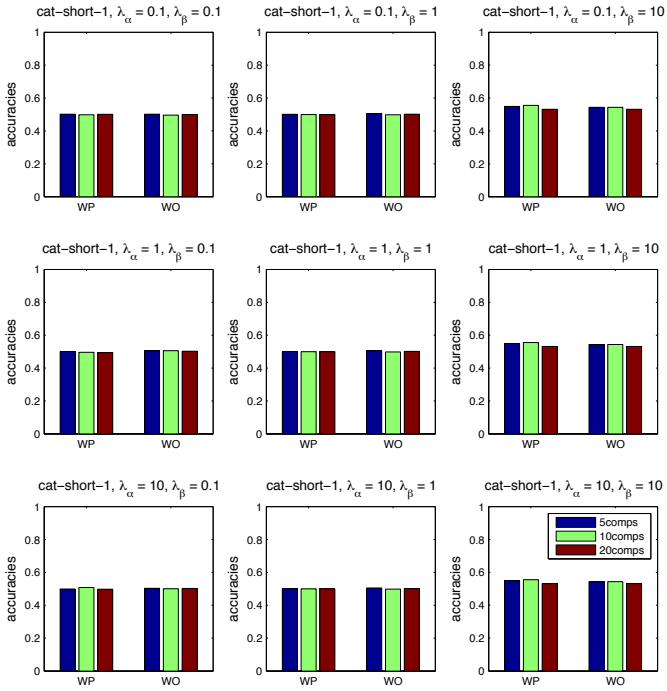


Figure A.1.2: The accuracies of the regularized bilinear regression when words from the cat-short-1 set are left out.

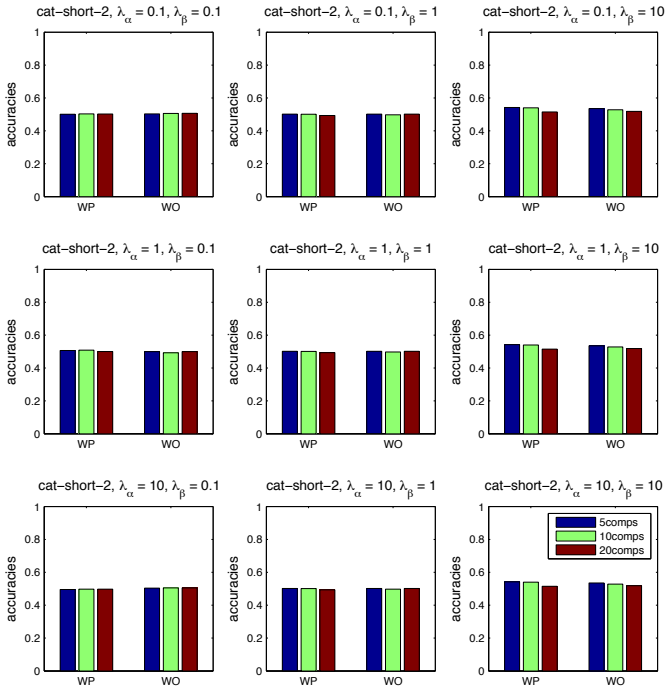


Figure A.1.3: The accuracies of the regularized bilinear regression when words from the cat-short-2 set are left out.

constraints assumed by the models we consider earlier, in particular, the constraints assumed by the earlier models in the first stage that learns the common features. One constraint imposed by the earlier models that use CCA to learn common features is to assume that the common feature scores each has length (or squared norm) one. No such constraint is imposed in the regularized bilinear regression. We investigate the length of each configuration. The results when we use all 60 words are shown in table A.1.1.

We can see in table A.1.1 that in most cases the average length of the factor scores does not exceed 0.1. There are a few exceptions, notably in the case $\lambda_\alpha = 10, \lambda_\beta = 1$, where the average length is around 60. However, besides this case, the exceptions occur when $\lambda_\beta = 10$. We see before that when $\lambda_\beta = 10$ the accuracies are higher compared to the other cases, and we see also that the length of the factor scores tend to be closer to one in this case. This suggests some relationship between the accuracies and the size of the factor scores.

What happens when there are missing words? The length of the factor scores for the various configurations are shown in tables A.1.2 (for the cat-short-1 case) and A.1.3 (for the cat-short-2 case). In these tables, we can first see that for a specific number of common features, there are no variations in the cases when $\lambda_\beta = 1$ and $\lambda_\beta = 10$. This can be considered as an extreme case of the more dominant effect of the λ_β parameter compared to the effect of the λ_α parameter, which we observe earlier in the accuracies,. Also, as we see in the case when all words are used, the length of the factor scores tends to be below 0.1 with the exception when $\lambda_\beta = 10$.

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0391 (0.0085)	0.0875 (0.0189)	0.0775 (0.0146)
$\lambda_\alpha = 1$	0.0385 (0.0036)	0.0631 (0.0202)	0.5437 (0.0959)
$\lambda_\alpha = 10$	0.0462 (0.0070)	0.0491 (0.0074)	0.0632 (0.0107)

(a) 5 common features

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0406 (0.0053)	0.0416 (0.0050)	0.5878 (0.1700)
$\lambda_\alpha = 1$	0.0413 (0.0024)	0.0386 (0.0053)	0.1314 (0.0601)
$\lambda_\alpha = 10$	0.0480 (0.0039)	0.0928 (0.0297)	0.0510 (0.0058)

(b) 10 common features

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0414 (0.0055)	0.0397 (0.0039)	0.0589 (0.0073)
$\lambda_\alpha = 1$	0.0407 (0.0053)	0.0403 (0.0055)	0.7738 (0.1937)
$\lambda_\alpha = 10$	0.0519 (0.0080)	60.5582 (9.3640)	0.1128 (0.0146)

(c) 20 common features

Table A.1.1: The average and the standard deviation (in parentheses) of the length of the factor scores when all the 60 words are used.

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0419 (0.0088)	0.0567 (0.0091)	0.0813 (0.0045)
$\lambda_\alpha = 1$	0.0482 (0.0099)	0.0567 (0.0091)	0.0813 (0.0045)
$\lambda_\alpha = 10$	0.0431 (0.000)	0.0567 (0.0091)	0.0813 (0.0045)

(a) 5 common features

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0458 (0.0057)	0.0637 (0.0132)	33.2991 (7.9899)
$\lambda_\alpha = 1$	0.0457 (0.0043)	0.0637 (0.0132)	33.2991 (7.9899)
$\lambda_\alpha = 10$	0.0505 (0.0127)	0.0637 (0.0132)	33.2991 (7.9899)

(b) 10 common features

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0459 (0.0064)	0.0471 (0.0063)	1.9151 (0.7979)
$\lambda_\alpha = 1$	0.0449 (0.0058)	0.0471 (0.0063)	1.9151 (0.7979)
$\lambda_\alpha = 10$	0.0596 (0.0083)	0.0471 (0.0063)	1.9151 (0.7979)

(c) 20 common features

Table A.1.2: The average and the standard deviation (in parentheses) of the length of the factor scores when words from the cat-short-1 set are left out.

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0423 (0.0036)	0.0576 (0.0122)	15.4550 (2.8061)
$\lambda_\alpha = 1$	0.0452 (0.0063)	0.0576 (0.0122)	15.4550 (2.8061)
$\lambda_\alpha = 10$	0.0450 (0.0050)	0.0576 (0.0122)	15.4550 (2.8061)

(a) 5 common features

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0448 (0.0070)	0.0442 (0.0048)	17.1549 (4.6802)
$\lambda_\alpha = 1$	0.0437 (0.0067)	0.0442 (0.0048)	17.1549 (4.6802)
$\lambda_\alpha = 10$	0.0487 (0.0075)	0.0442 (0.0048)	17.1549 (4.6802)

(b) 10 common features

	$\lambda_\beta = 0.1$	$\lambda_\beta = 1$	$\lambda_\beta = 10$
$\lambda_\alpha = 0.1$	0.0457 (0.0046)	0.0490 (0.0081)	0.1394 (0.0459)
$\lambda_\alpha = 1$	0.0456 (0.0052)	0.0490 (0.0081)	0.1394 (0.0459)
$\lambda_\alpha = 10$	0.0443 (0.0033)	0.0490 (0.0081)	0.1394 (0.0459)

(c) 20 common features

Table A.1.3: The average and the standard deviation (in parentheses) of the length of the factor scores when words from the cat-short-2 set are left out.

A.2 The Conditional Factor Analysis Model

Another way to estimate the parameters in equations (A.1) and (A.2) is to use the conditional factor analysis model (`http://learning.eng.cam.ac.uk/zoubin/software/cfa.tgz`, Ghahramani and Hinton (1996)). More precisely, this model gives a probabilistic formulation of equations (A.1) and (A.2) when there is one group ($M = 1$):

$$\mathbf{g}_w \sim \mathcal{N}(\mathbf{A}\mathbf{f}_w, \mathbf{Q}) \tag{A.13}$$

$$\mathbf{y}_w \sim \mathcal{N}(\mathbf{B}\mathbf{g}_w, \mathbf{R}) \tag{A.14}$$

for some diagonal covariance matrices $\mathbf{Q}$ ($K \times K$) and $\mathbf{R}$ ($D \times D$). An EM algorithm can be derived to obtain the maximum-likelihood estimates for the parameters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{Q}$, and $\mathbf{R}$.

As mentioned above, as originally formulated the model assumes $M = 1$. The extension to the case when M is arbitrary is straightforward. Letting

$$\mathbf{y}_w = \begin{bmatrix} \mathbf{y}_w^{(1)} \\ \vdots \\ \mathbf{y}_w^{(M)} \end{bmatrix} \tag{A.15}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{B}^{(M)} \end{bmatrix}, \tag{A.16}$$

we can use the EM algorithm for the case $M = 1$ to obtain maximum-likelihood estimates when the number of groups M is arbitrary.

A.2.1 Experimental results

We perform experiments with the conditional factor analysis model like the ones described in section 4.2. Here we consider only when we have the subjects from the WP study (9 subjects), and we try the conditional factor analysis model with 10, 20, and 30 factors. The accuracies are shown in figure A.2.1.

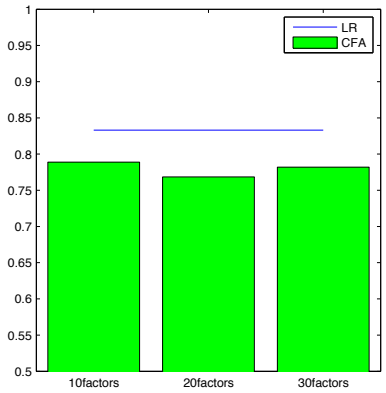


Figure A.2.1: The accuracies of the conditional factor analysis method with 10, 20, and 30 factors, shown in the green bars. The blue line shows the accuracy of the baseline LR method.

In this figure, we also show the accuracies when applying the (single-subject) baseline LR model. As we can see in the figure, for all three cases, the accuracies of the conditional factor analysis model are significantly lower compared to the accuracy of the baseline LR model for this task, which in turn is significantly lower compared to the accuracy of the best two-stage method in chapter 4.

A.3 Discussion

All the accuracies of the variations of the regularized bilinear regression and the conditional factor analysis models are lower compared to the accuracies of the two-stage methods we consider earlier in the chapter. However, these two models might still have potential. Given what we show in this appendix, for the regularized bilinear regression model, a constraint on the length of the factor scores might lead to better accuracies. Whether this can be done using the regularization we present here, or whether more explicit constraints are needed, requires more investigation. For the conditional factor analysis model, better accuracies can potentially be obtained when we incorporate stronger prior distributions on the parameters. Given that we have few instances compared to the number of variables in the model, this kind of prior distributions can help significantly in guiding the estimation procedure to the more appropriate parameter space. We leave a more extensive exploration of these two models as part of future work.

Appendix B

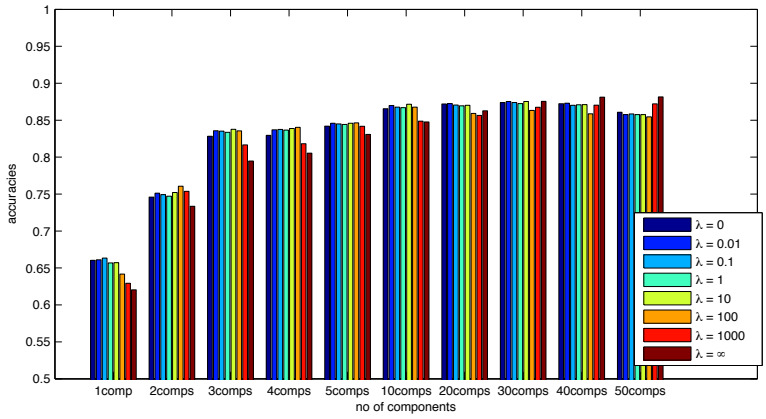
Sensitivity to the Regularization Parameter When Applying Canonical Correlation Analysis

Here we briefly explore how sensitive the experimental results when we use canonical correlation analysis (CCA) are with respect to the regularization parameter used. In particular, we consider the experiments performed in section 4.2 and try the set of values $\{0, 0.01, 0.1, 1, 10, 100, 1000, \infty\}$ for λ, where $\lambda = \frac{\kappa}{1-\kappa}$ and κ is the regularization parameter for CCA as described in chapter 3. In other words, we convert κ to something that ranges from 0 to ∞. Note that the setting $\kappa = 0.5$ that we use in the original experiments is equivalent to $\lambda = 1$. We focus on the intel218 features. The results of the different settings of λ are shown in figure B.1 for the CCA-mult method, and in figure B.2 for the CCA-mult-comb method.

In these figures, we see that in general the trends that we see in figure 4.2.2 still persist for most settings of λ, in the sense that the accuracies are low when the number of components is low and they increase as the number of components increases. Note that in the original experimental setting using $\kappa = 0.5$ (equivalent to $\lambda = 1$), the best accuracies are obtained when use around 20-30 components. However, with the settings considered here we see that we can obtain better accuracies when we use $\lambda = \infty$; in this setting ($\lambda = \infty$), the best accuracies are obtained when we use 50 components.

Given these findings, it would be interesting to explore different regularization parameter settings for the other experiments that we run. In addition, We can also explore how the loadings and scores vary when we vary the regularization parameters. Note that these results are based on setting the same regularization parameter for all the datasets. It is possible to explore the sensitivity of the results when we vary independently the regularization parameter for each dataset.

(a) CCA-mult-WP



(b) CCA-mult-WO

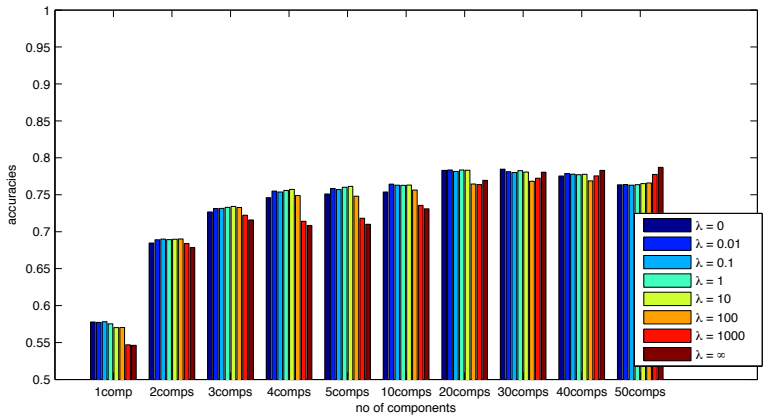
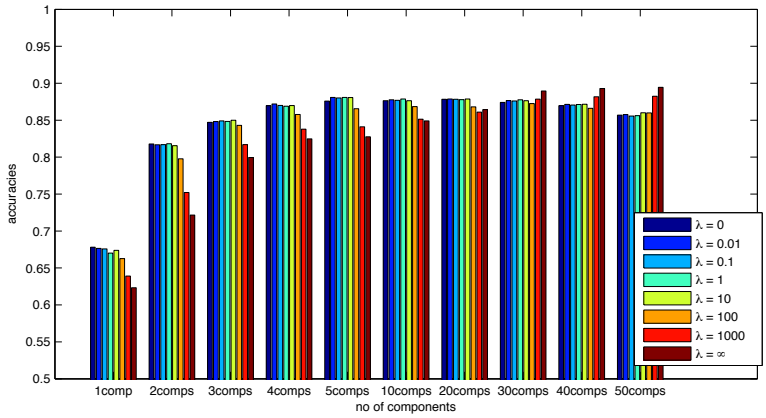


Figure B.1: Accuracies of the CCA-mult methods with different settings of the parameter λ, as a function of the number of components

(a) CCA-mult–comb-WP



(b) CCA-mult-comb-WO

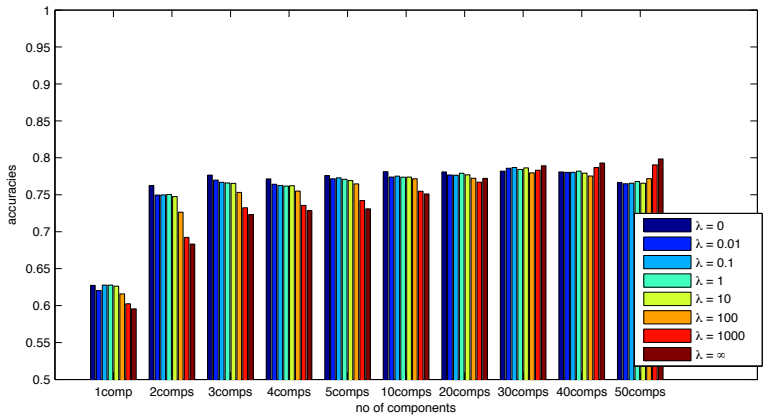


Figure B.2: Accuracies of the CCA-mult-comb methods with different settings of the parameter λ, as a function of the number of components

Appendix C

ROI Glossary

ROI	description
(L/R)PRECENT	left/right precentral
(L/R)SUPFRONT	left/right superior frontal
(L/R)ORBFRONT	left/right orbitofrontal
(L/R)MIDFRONT	left/right mid frontal
(L/R)OPER	left/right pars opercularis
(L/R)TRIA	left/right pars triangularis
(L/R)INSULA	left/right insula
(L/R)SMA	left/right supplementary motor area
(L/R)MEDFRONT	left/right medial frontal
(L/R)ACING	left/right anterior cingulate
(L/R)PCING	left/right posterior cingulate
(L/R)HIP	left/right hippocampus
(L/R)PARAHIP	left/right parahippocampal
(L/R)AMYG	left/right amygdala
(L/R)CALC	left/right calcarine
(L/R)SES	left/right superior extrastriate
(L/R)IES	left/right inferior extrastriate
(L/R)FUSIFORM	left/right fusiform
(L/R)POSTCENT	left/right postcentral
(L/R)SPL	left/right superior parietal
(L/R)IPL	left/right inferior parietal
(L/R)IPS	left/right intraparietal sulcus
(L/R)CAUDATE	left/right caudate
(L/R)PUTAMEN	left/right putamen
(L/R)PALLIDUM	left/right pallidum
(L/R)THALAMUS	left thalamus
(L/R)HESCHL	left/right Heschl
(L/R)TPOLE	left/right temporal pole
(L/R)STANT	left/right anterior superior temporal
(L/R)STMID	left/right mid superior temporal
(L/R)STPOS	left/right posterior superior temporal
(L/R)ITMID	left/right mid inferior temporal
(L/R)ITPOS	left/right posterior inferior temporal
(L/R)CBEL	left/right cerebellum
CBELVERMIS	cerebellum vermis
(L/R)ITANT	left/right anterior inferior temporal

Bibliography

Alter, O., Brown, P. O., and Botstein, D. (2003). Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proceedings of the National Academy of Sciences*, 100(6):3351–3356.

Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48.

Bach, F. R. and Jordan, M. I. (2005). A Probabilistic Interpretation of Canonical Correlation Analysis. Technical report, Department of Statistics, University of California, Berkeley.

Bakker, B. and Heskes, T. (2003). Task Clustering and Gating for Bayesian Multitask Learning. *Journal of Machine Learning Research*, 4:83–99.

Beckmann, C. and Smith, S. (2005). Tensorial extensions of independent component analysis for multisubject FMRI analysis. *NeuroImage*, 25(1):294–311.

Beckmann, C. F. and Smith, S. M. (2004). Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, 23(2):137–152.

Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28(1):41–70.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*. Duxbury, 2nd edition.

Casey, B., Cohen, J. D., O'Craven, K., Davidson, R. J., Irwin, W., Nelson, C. A., Noll, D. C., Hu, X., Lowe, M. J., Rosen, B. R., Truwitt, C. L., and Turski, P. A. (1998). Reproducibility of fMRI Results across Four Institutions Using a Spatial Working Memory Task. *NeuroImage*, 8:249–261.

Daumé III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26:101–126.

Davatzikos, C., Ruparel, K., Fan, Y., Shen, D., Acharyya, M., Loughead, J., Gur, R., and Langleben, D. (2005). Classifying spatial patterns of brain activity with machine learning methods: Application to lie detection. *NeuroImage*, 28:663–668.

Demidenko, E. (2004). *Mixed Models: Theory and Applications*. Wiley-Interscience.

Demmel, J. W. (1997). *Applied Numerical Linear Algebra*. SIAM.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39:1–38.

Eddy, W. F., Fitzgerald, M., Genovese, C. R., Mockus, A., and Noll, D. C. (1996). Functional image analysis software — computational olio. In Prat, A., editor, *Proceedings in computational statistics*, pages 39–49.

Efron, B. (1982). *The Jackknife, the Bootstrap, and Other Resampling Plans*. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM.

Evans, A. C., Collins, D. L., Mills, S. R., Brown, E. D., Kelly, R. L., and Peters, T. M. (1993). 3D statistical neuroanatomical models from 305 MRI volumes. In *Proc. IEEE-Nuclear Science Symposium and Medical Imaging Conference*.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977). An algorithm for finding best matches in logarithmic

expected time. *ACM Transactions on Mathematical Software*, 3(3):209–226.

Friston, K. J., Glaser, D. E., Henson, R. N. A., Kiebel, S., Phillips, C., and Ashburner, J. (2002a). Classical and Bayesian inference in neuroimaging: Applications. *NeuroImage*, 16:484–512.

Friston, K. J., Penny, W., Phillips, C., Kiebel, S., Hinton, G., and Ashburner, J. (2002b). Classical and Bayesian inference in neuroimaging: Theory. *NeuroImage*, 16:465–483.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1(3):515–534.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2003). *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition.

Ghahramani, Z. and Hinton, G. E. (1996). The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto.

Griffiths, T. and Ghahramani, Z. (2006). Infinite latent feature models and the Indian buffet process. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*. MIT Press.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16:2639–2664.

Harville, D. (1977). Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, 72:320–338.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, 2nd edition.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(7):498–520.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3/4):321–377.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. Wiley-Interscience.

Johnson, R. A. and Wichern, D. W. (2002). *Applied Multivariate Statistical Analysis*. Prentice Hall, 5th edition.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37:183–233.

Just, M. A., Cherkassky, V. L., Aryal, S., and Mitchell, T. M. (2010). A neurosemantic theory of concrete noun representation based on the underlying brain codes. *PLoS ONE*, 5(1).

Kaiser, H. F. (1957). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200.

Kettenring, J. R. (1971). Canonical analysis of several sets of variables. *Biometrika*, 58(3):433–451.

Kiebel, S. and Holmes, A. (2003). The general linear model. In Frackowiak, R., Friston, K., Frith, C., Dolan, R., Friston, K., Price, C., Zeki, S., Ashburner, J., and Penny, W., editors, *Human Brain Function*. Academic Press.

Lazar, N. A., Luna, B., Sweeney, J. A., and Eddy, W. F. (2002). Combining brains: A survey of methods for statistical pooling of information. *NeuroImage*, 16:538–550.

Logothetis, N. K., Pauls, J., Augath, M., Trinath, T., and Oeltermann, A. (2001). Neurophysiological investigation of the basis of the fMRI signal. *Nature*, 412:150–157.

Marx, Z., Rosenstein, M. T., Kaelbling, L. P., and Dietterich, T. G. (2005). Transfer Learning with an Ensemble of Background Tasks. In *Inductive Transfer: 10 Years Later, NIPS Workshop*.

McKeown, M., Makeig, S., Brown, G., Jung, T., Kindermann, S., Bell, A., and Sejnowski, T. (1998). Analysis of fMRI data by blind separation into independent spatial components. *Human Brain Mapping*, 6(3):160–188.

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., and Newman, S. (2004).

Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175.

Mitchell, T. M., Shinkareva, S. V., Carlson, A., Chang, K.-M., Malave, V. L., Mason, R. A., and Just, M. A. (2008). Predicting human brain activity associated with the meanings of nouns. *Science*, 320:1191–1195.

Morris, C. N. (1983). Parametric Empirical Bayes Inference: Theory and Applications. *Journal of the American Statistical Association*, 78(381):47–55.

Mourão-Miranda, J., Reynaud, E., McGlone, F., Calvert, G., and Brammer, M. (2006). The impact of temporal compression and space selection on SVM analysis of single-subject and multi-subject fMRI data. *NeuroImage*, 33:1055–1065.

Paige, C. C. and Saunders, M. A. (1981). Towards a generalized singular value decomposition. *SIAM Journal of Numerical Analysis*, 18(3):398–405.

Palatucci, M., Pomerleau, D., Hinton, G., and Mitchell, T. M. (2010). Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems 22*.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572.

Penny, W., Holmes, A., and Friston, K. (2003). Random effects analysis. In Frackowiak, R., Friston, K., Frith, C., Dolan, R., Friston, K., Price, C., Zeki, S., Ashburner, J., and Penny, W., editors, *Human Brain Function*. Academic Press, 2nd edition.

Pereira, F., Mason, R., Mitchell, T., Just, M., and Kriegeskorte, N. (2006). Decoding of semantic category information from single trial fMRI activation in response to word stimuli, using searchlight voxel selection. In *12th Conference on Human Brain Mapping*.

Ponnapalli, S. P., Saunders, M. A., Golub, G. H., and Alter, O. (2009). A higher-order generalized singular value decomposition for comparative analysis of large-scale datasets. Under revision.

Raudenbush, S. W. and Bryk, A. S. (2001). *Hierarchical Linear Models: Applications and Data Analysis Methods*. Sage.

Reichle, E. D., Carpenter, P. A., and Just, M. A. (2000). The neural bases of strategy and skill in sentence-picture verification. *Cognitive Psychology*, 40:261–295.

Rosenstein, M. T., Marx, Z., Kaelbling, L. P., and Dietterich, T. G. (2005). To Transfer or Not To Transfer. In *Inductive Transfer: 10 Years Later, NIPS Workshop*.

Roweis, S. (1998). EM algorithms for PCA and SPCA. In *Advances in Neural Information Processing Systems 10*.

Roweis, S. and Ghahramani, Z. (1999). A unifying review of linear Gaussian models. *Neural Computation*, 11:305–345.

Roy, D. M. and Kaelbling, L. P. (2007). Efficient Bayesian Task-Level Transfer Learning. In *Proceedings of the 20th Joint Conference on Artificial Intelligence*.

Saito, N. and Coifman, R. R. (1995). Local discriminant bases and their applications. *Journal of Mathematical Imaging and Vision*, 5:337–358.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

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

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622.

Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., and Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525.

Van Loan, C. F. (1976). Generalizing the singular value decomposition. *SIAM Journal of Numerical Analysis*,

13(1):76–83.

Vinod, H. D. (1976). Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4:147–166.

Wang, X., Hutchinson, R., and Mitchell, T. M. (2004). Training fMRI classifiers to discriminate cognitive states across multiple subjects. In *NIPS*.

Wei, X., Yoo, S.-S., Dickey, C. C., Zou, K. H., Guttmann, C. R., and Panych, L. P. (2004). Functional MRI of auditory verbal working memory: long-term reproducibility analysis. *NeuroImage*, 21:1000–1008.

West, M. (2003). Bayesian factor regression models in the "large p, small n" paradigm. In Bernardo, J., Bayarri, M., Berger, J., Dawid, A., Heckerman, D., Smith, A., and West, M., editors, *Bayesian Statistics 7*, pages 733–742. Oxford University Press.

Woods, R. P. (1996). Modeling for intergroup comparisons of imaging data. *NeuroImage*, 4:S84–S94.

Xue, Y., Liao, X., Carin, L., and Krishnapuram, B. (2007). Multi-Task Learning for Classification with Dirichlet Process Priors. *Journal of Machine Learning Research*, 8:35–63.

Yin, X. (2004). Canonical correlation analysis based on information theory. *Journal of Multivariate Analysis*, 91:161–176.

Yu, K., Tresp, V., and Schwaighofer, A. (2005). Learning Gaussian Processes from Multiple Tasks. In *Proceedings of the 22nd International Conference on Machine Learning*.

Zhang, J., Ghahramani, Z., and Yang, Y. (2006). Learning Multiple Related Tasks using Latent Independent Component Analysis. In Weiss, Y., Schölkopf, B., and Platt, J., editors, *Advances in Neural Information Processing Systems 18*, pages 1585–1592. MIT Press, Cambridge, MA.