FEATURE EXTRACTION AND MATCHING IN CONTENT-BASED RETRIEVAL OF FUNCTIONAL MAGNETIC RESONANCE IMAGES

BY BING BAI

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Paul Kantor

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE DISSERTATION

Feature extraction and matching in content-based retrieval of functional Magnetic Resonance Images

by Bing Bai

Dissertation Director: Paul Kantor

Functional Magnetic Resonance Imaging (fMRI) has become a widely used technique in neuroscience research. Brain regions corresponding to certain cognitive functionalities can be located by studying the intensity change in a series of 3D brain scans.

Although fMRI has been widely studied, little attention has been paid to content-based ("content" means the explicit or implicit cognitive process) retrieval of images despite the existence of databases equipped with textual description (fMRIDC). Content-based retrieval is potentially useful in discovering brain activation patterns, and in diagnoses by comparing observed patterns with those of known diseases, leading to clinical applications.

We conducted a comprehensive investigation of feature extraction and similarity measures used in several research communities (including information retrieval

(IR), signal processing, and computer vision(CV)), to set up a content-based retrieval framework for a large, heterogeneous database. We developed methods for both hypothesis-based (stimulus known) and hypothesis-free (stimulus unknown) schemes. For the former, we adapted and extended an adaptive Finite Impulse Response (FIR) Model to get a more robust estimation of the activation level of brain regions. We then relaxed the assumption that the brain responds as a linear time-invariant (LTI) system, by using a 4-parameter ordinary differential equation to model brain responses. We then evaluated a number of similarity measures used in IR and CV, such as Latent Semantic Indexing (LSI), TFIDF, and Mahalanobis distance, etc. For the latter, we used a heuristic to select independent components with low mean temporal frequency, and applied a maximum weight bipartite matching technique to integrate component-level similarity and give a more robust retrieval performance.

For feature selection, we found that an FIR model with a smoothing factor can improve retrieval performance significantly. For feature matching, a method similar to "dilation operators" used in image processing gives better and more robust retrieval performance than other methods.

Acknowledgements

First, I want to thank my Ph.D advisor, Dr. Paul B. Kantor. I am fortunate to have had him as my advisor. He initiated many of the ideas in this study, and provided insights from a broad and deep understanding of many problems. More importantly, he has always been available to help me through this long research process, and he helps in a way to overcome my anxiety.

I want to thank my lab mates Dr. Nicu Cornea and Dr. Ulukbek Ibraev who helped me a lot. I still rely on many tools they developed. Many thanks to other faculty members in our research group: Dr. Sven Dickinson, Dr. Deborah Silver, and especially Dr. Ali Shokoufandeh. I learned so much from them. Thanks also to Dr. Vladimir Pavlovic and Dr. Michael Littman for their guidance in some research problems.

I like to thank my friend Dr. Robert Rittman. Not only because he helped edit my dissertation, but because of his kind and heart warming help when I was experiencing a difficult time.

Finally, I want to thank my girlfriend, Yihua Wu, and my sister Dr. Jie Bai. Their constant support and encouragement enabled me to finish this work.

Dedication

To my parents: Shangrong Bai and Guilian Li. Without their selfless love, I could not achieve anything.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

The goal of our research is to design efficient and effective information retrieval algorithms for functional Magnetic Resonance Images (fMRI) [Ogawa et al., 1992, Kwong et al., 1992, Frackowiak et al., 2004].

Briefly speaking, fMRI is a technique used to monitor brain activity. The most widely used fMRI method is BOLD (blood oxygen level dependent). The BOLD technique is effective because the intensity of an image element (it is called a *voxel* in 3D images, corresponding to a "pixel" for 2D images) is related to the level of blood oxygen in the corresponding brain region. When a cognitive process involves a specific brain region, at first, some oxygen is consumed, but more oxygen is brought in by blood flow soon after, and this change will brighten the brain region in the image (as shown in Figure 1.1). In a typical fMRI experiment, the experimental subject is assigned a certain type of cognitive task (which is referred to here as the *stimulus*) at scheduled times. Between tasks, the subject focuses on something different than the assigned task - something relatively undemanding, such as watching a cross hair. We call the former status the *condition*, and the latter status the *control*. A 3D brain scan is made every few seconds, during each experimental "run" session. By comparing the intensities of voxels during the condition and the control periods, we can estimate which brain regions are "activated" by the cognitive process.

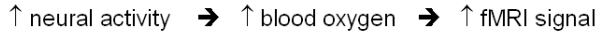


Figure 1.1: Neural activities cause stronger (brighter) fMRI signals

As a method for watching "how the brain works", fMRI has been used as a powerful research tool for many aspects of neuroscience studies in the past decade. More recently, fMRI is gaining clinical attention. For example, studies [Thulborn et al., 2000] of some Alzheimer's disease patients show that differences between their brains and typical brains can be detected using scans before the symptoms are outwardly apparent. [Ford et al., 2003] classifies brains with different diseases from normal brains.

Meanwhile, many fMRI experiments have been conducted, and we expect many more of them in future. As the size of the (world) data corpus increases, efficient data *sharing* schemes for fMRI data become more and more desirable. Data sharing, of course, provides a larger database for testing and validating analytical algorithms. More importantly, different researchers may find different values in the same data, discovering similarities in the brain's activity, when the cognitive tasks do not seem to be related, based on psychological reasoning alone. With this idea of sharing in mind, an online fMRI library – the Dartmouth fMRI data center (fMRIDC) [Horn et al., 2001] – was founded in 2003. As of May 2006, the fMRIDC has archived brain images from 120 experiments and has received 2000 data requests. This valuable data is presently indexed with textual descriptions (also called "textual meta-data"), including the technical configurations under which the images were collected, subject information, information about the cognitive tasks (provided via related publications) and any other information the providers consider important.

While fMRIDC is a big step in boosting fMRI research, it has a drawback: there is not an efficient way to relate different experiments. Different cognitive tasks may share some latent neural processes (such as a visual process), and thus have some level of similarity. However, it is difficult to describe this similarity using the current fMRIDC system.

In this thesis, we report investigations of *content-based* indexing and retrieval

of fMRI images. Our work is motivated by the potential for success, but we are also cognizant of the many challenges. In short, using a brain image as *query*, the system we are proposing should return images that involve the same or similar cognitive processes. The potential applications include, but are not limited to, the following:

- Helping researchers find similar studies and related research work.
- Helping researchers discover hidden similarities among superficially different studies.
- Helping doctors diagnose brain disorders, by looking at the clinical history of persons with similar fMRI patterns.

The study is performed in the framework of information retrieval (IR) [Salton and Buckley, 1988]. IR is most widely known in applications such as search engines, which usually have a huge database of documents and images. Similar to classification, IR describes the dataset with features. However, the IR framework is usually built to retrieve similar datasets from a very large database, in which it is difficult to assign class labels to each dataset. IR techniques usually use predefined "distances" to measure the similarity between the query and each dataset in the database, rather than "class labels", as in classification framework. This makes the IR framework more extensible since no explicit classifier is needed, and thus it is preferable to large databases from miscellaneous sources.

One straightforward idea is to map the brain image retrieval problem to the framework of content-based image [Veltkamp and Tanase, 2000] or video retrieval [Yoshitaka and Ichikawa, 1999]. However, common features used in the image retrieval community – such as colors, textures, and shapes – can not be applied to fMRI data directly. In particular, all fMRI images have similar brain structures, similar shapes and similar color/intensity ranges. Thus, in indexing fMRI images,

we are looking for very subtle intensity changes over space or time, which differs from the goal of most image/video retrieval algorithms.

Thus, we are looking for retrieval schemes that are appropriate for this special type of data. In this thesis, we extend the framework established in [Ibraev, 2005]. Specifically, we investigate feature extraction and feature matching, which are the two key problems in this framework. Our contributions can be summarized as the following:

1. We propose a shape-based Finite Impulse Response (FIR) model that produces more accurate the Hemodynamic Response Function (HRF) estimates. With these estimate we are able to generate better estimates of the activation map, thus potentially reaching a more stable performance in retrieval. (Chapter 5)

2. We develop a method to adapt FIR model for multiple conditions, with which FIR model can be fully enabled to do activation assessment in the framework of General Linear Model. (Chapter 5)

3. We propose a non-linear hemodynamic model that generates parameters with clearer physiological meaning. Nonlinearity of hemodynamic behavior has been a hot topic in recent fMRI research. The nonlinear model we present, with a relatively simpler form and fast resolution, achieves comparable performance in retrieval. (Chapter 5)

4. We explore retrieval models without stimulus information, based on Independent Component Analysis (ICA) components. This model is potentially useful in experiments where the stimulus information is not available or is not well defined. (Chapters 7 and 9)

5. We fit the brain activated regions to the textual retrieval framework. We adapt classical information retrieval techniques, such as inverted indexing,

TFIDF [Salton and Buckley, 1988] and latent semantic analysis (LSI), into fMRI indexing and retrieval. (Chapter 8)

This dissertation is organized into 4 major parts. In chapters 1-3, we review the background of fMRI technology, including the physical facts and the analytical methods such as GLM and ICA. In chapters 4-7, we will examine several models, including nonlinear model and FIR models with various forms, for activation mapping features for fMRI data. In chapters 8-9, we review several feature matching algorithms, including the algorithms we adapted from information and computer vision areas and the algorithm we developed. In chapters 10-15, we evaluate the performance of our proposed algorithms then give conclusions and discussions.

Note: This chapter includes parts of [Bai et al., 2007c], which are co-authored by Bing Bai, Paul Kantor, Nicu Cornea, and Deborah Silver.

Chapter 2
MRI and fMRI: principles

Magnetic Resonance Imaging (MRI) [Damadian, 1971, Lauterbur, 1973] is a widely used non-intrusive medical imaging technology. MRI is superior to other imaging systems such as CT (Computerized Tomography) or PET (Positron Emission Tomography) in many aspects. MRI usually has higher spatial resolution. It is less risky since MRI does not use X-rays or radioactive materials. Also, the subject does not need special preparation for an MRI exam which makes it more convenient.

fMRI [Kwong et al., 1992, Ogawa et al., 1992] is an application of MRI with more interest for neural scientists and psychologists. fMRI detects the neural activity of the brain instead of the physical structures of the brain.

2.1 MRI technology

2.1.1 Nuclear Magnetic Resonance (NMR)

The physical fundation of MRI is the NMR phenomenon described by [Bloch et al., 1946, Purcell et al., 1946]. NMR is closely related with so-called "spin physics", which studies the spin theory of basic particles like protons and neutrons. Knowledge of quantum mechanics is required to thoroughly understand NMR. However, an approximate model in a classical mechanical framework is commonly used to demonstrate the principle of NMR. We will introduce this model briefly.

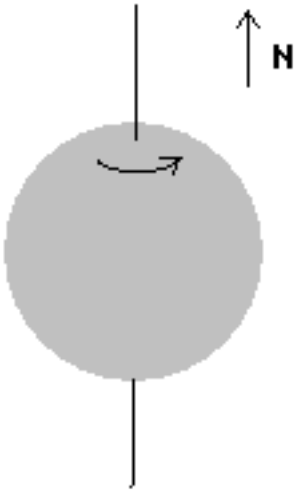


Figure 2.1: The magnetic field generated by a spinning proton

Spin of proton and its magnetic field.

In this model, a proton is considered as a solid object, which is always spinning on its axis. Since protons carry a charge, this spin produces a magnetic field by the right hand rule, as shown in Figure 2.1.

When protons are put in a strong external magnetic field, their spinning directions are aligned parallel to the external magnetic field. According to classical electromagnetics, the direction of the magnetic field b of the proton should be opposite to the direction of the external magnetic field B_0 to minimize the potential energy. We call this the "low energy state". However, a proton can jump to a "high energy state" from a "low energy state" by absorbing a photon, or jump back to a "low energy state" from a "high energy state" by releasing a photon. In the "high energy state", the direction of the micro magnetic field is the same as the external magnetic field as shown in Figure 2.2.

In these transition processes, the energy of the photon must match exactly the energy difference between two energy levels, which depends on the strength

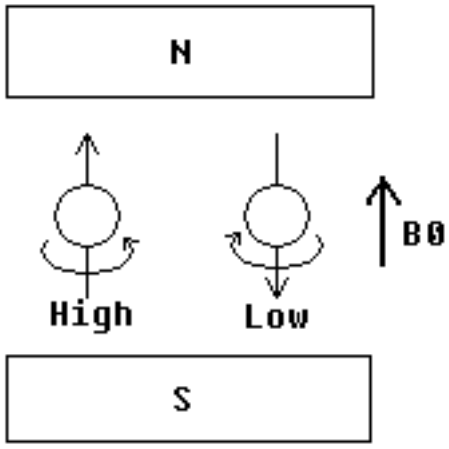


Figure 2.2: The spinning protons in an external magnetic field B0. The left is at high energy state, while the right is at low energy state.

of the external magnetic field and the gyromagnetic ratio γ of the nucleus. The frequency of the photon can be described by following equation:

$$v = \gamma B \tag{2.1}$$

In Eq.2.1, γ is called gyromagnetic ratio, and is dependent on the type of nucleus. B is the strength of external magnetic field. The energy of the photon is:

$$E = hv \tag{2.2}$$

and $h = 6.626 \times 10^{-34} J \cdot s$ is Planck constant. Table 2.1.1 shows the γ value of different nuclei.

In MRI, the photons are in the frequency range of radio frequency, thus we use RF to identify the electromagnetic stimulation of nuclear spin.

Spin Packets.

In our classical mechanics model, we divide the microscopic spins into groups called "spin packets". A spin packet is a small cell in space in which the external

Nuclei	*MHz/T*
^{1}H	42.58
^{2}H	6.54
^{31}P	17.25
^{23}Na	11.27
^{14}N	3.08
^{13}C	10.71
$^{1}9F$	40.08

Table 2.1: Gyromagnetic ratios of different atoms

magnetic field is essentially constant for all atoms in it (later we will see that we manage to let spin packets have different strength of magnetic field). We consider the spins in this spin packet as one single spin. The sum of magnetic fields of all spin packets is called net magnetization. The net magnetization is called the equilibrium magnetization in B_0 if there is no radio frequency input.

Spin Relaxation

To simplify our description, we assume the external magnetic field B_0 is parallel to z axis.

The magnetic field of protons is realigned when the RF input is turned on. When the outside electromagnetic wave is turned off, the high energy protons release photons, and jump back to the low energy state. This procedure is called spin relaxation. In a macroscopic view, a spin relaxation consists of two processes. The first process is called "spin-lattice relaxation", which describes the net magnetization gradually returning to the equilibrium magnetization in the direction of B_0. A time parameter T1 is used to indicate how slowly this magnetic field returns to the equilibrium position.

The second process is more interesting. Suppose the net magnetic field M is not parallel to B_0 (i.e. z axis in this case), then the movement of M is a rotation

around B_0, namely precession. Precession is mostly discussed in classical mechanics as the motion of a top. Suppose a rotating top has an angular momentum of $\vec{L}$, and a torque $\vec{\tau}$ is applied, then the change of this rotation can be described as in Eq. 2.3.

$$\frac{d\vec{L}}{dt} = \vec{\tau} \tag{2.3}$$

See Figure 2.3 for a illustration.

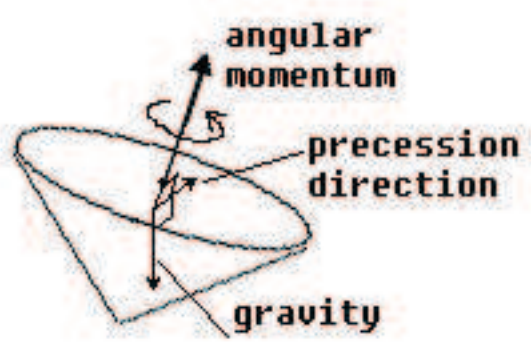


Figure 2.3: Precession of a rotating top. The precession is in the direction of cross product of angular momentum and the gravity

The precession of net magnetization has the same frequency as the photon frequency in equation 2.1. This frequency is also called the Larmor frequency. This fact explains the word "Resonance" in NMR. Similar to other resonance phenomena, only a stimulus with the same frequency as the internal frequency of an object can increase the potential energy of the object. Here the potential energy is labeled by the spin direction which is directly related to the high/low energy states.

Generating Electric Signal-Based on Precession.

Precession makes the magnetization rotate around the z-axis. If we put a coil close enough in a plane parallel to the z-axis (as shown in figure 2.4 (a)), an alternating

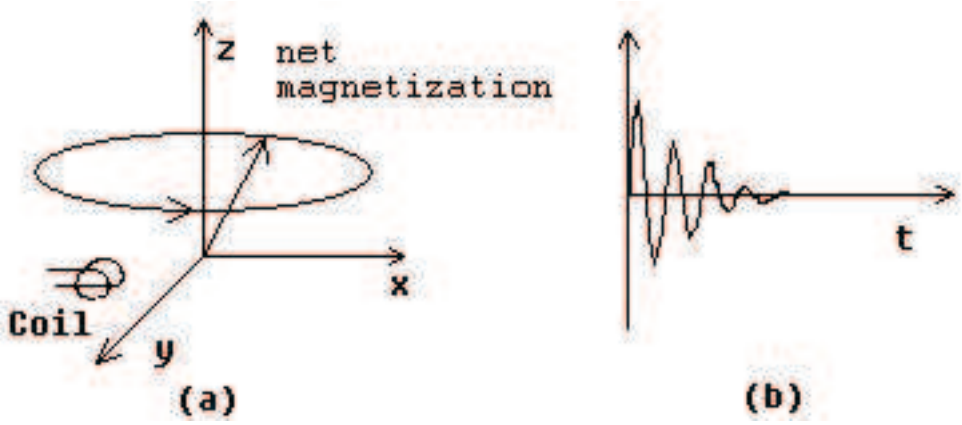


Figure 2.4: Precession of net magnetization generates current in nearby coils. (a) A coil close to precessing magnetization. (b) The alternating current in the coil

current is generated in the coil because of the variation of the magnetic field. The precession decays very quickly, for reasons we skip here, so the current decays quickly as well (see 2.4 (b)).

Pulse Sequence

As we stated earlier, the equilibrium magnetization is in the z-axis direction. To allow precession to happen, we have to move the net magnetization to make a component not on the z-axis. The idea is still precession, but under another external magnetic field B_1. The direction of B_1 should always be perpendicular to current net magnetization and the z-axis, so the net magnetization can rotate down to the x-y plane. Note that once the net magnetization leaves the z-axis, it also has precession due to B_0, so the direction of B_1 also rotates around z-axis in Larmor frequency. This B_1 can be generated by a coil with alternating current. We can control the degree of net magnetization by the duration of the current.

2.1.2 Imaging Principles

We have introduced the physical basis for MRI, but we still have not shown how the images are generated. This will be covered in this section. We need to remember the following two facts:

- A resonance signal happens only if the RF stimulus is at the Larmor frequency, which depends on both the external magnetic field and the type of atom.
- The strength of a resonance signal is proportional to the number of atoms whose Larmor frequency is RF.

Gradient magnetization.

One way to encode space is by using a magnetic field gradient. In a homogeneous magnetic field, we can get a signal with a certain RF, but we can not know the contribution from each part of the subject. In other words, we can not get any position information from this signal. To overcome this problem, we have to encode the location into the signal in some way. This is done with a magnetic field gradient as shown in Figure 2.5.

In 2.5 (a), the strength of the magnetic field is increasing from left to the right. Positions with different x coordinates will be in different magnetic fields, which makes them respond to different RF. To be more precise, the Larmor frequency is proportionate to the offset of the atoms. This is called *frequency encoding*.

We use an example of only 3 resonance resources to illustrate the idea. In Figure 2.5 (b), we see that the point A has a resonance at frequency f_A, and points B and C cause a *bigger* resonance at frequency f_B. The amplitude of the signal is proportional to the number of points at the corresponding location.

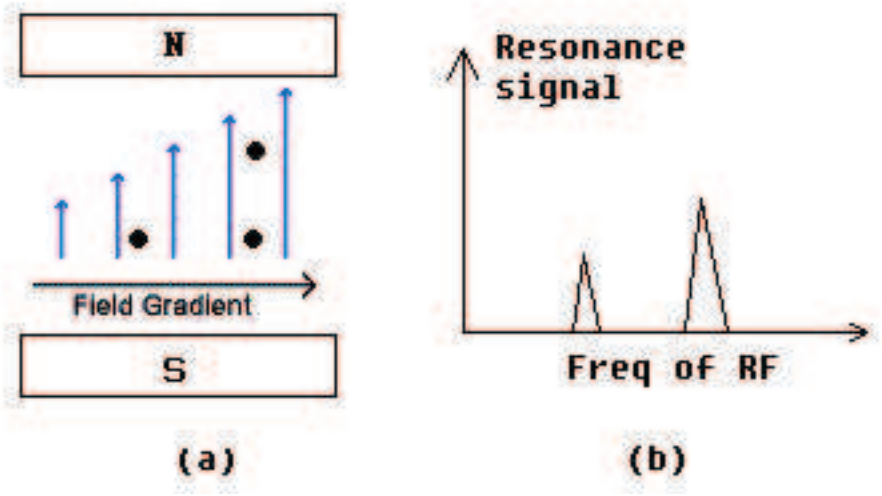


Figure 2.5: From [Hornak, 2005]. (a) The magnetic field gradient, the strength is proportional to x coordinates, the points are the sources of resonance signals. (b) The resonance signals. The strength of signal at certain frequency is proportional to number of resonance sources at corresponding x coordinates.

Back projection method.

Back projection is one of the oldest methods used in MRI [Lauterbur, 1973] and is based on frequency encoding we just described. It is a very good method to show how MRI images are built. In Figure 2.6, we cast B_0 in different directions, and get different signals in these directions. If we project the signal in the direction of B_0, all the pixels in the projection will get an intensity proportional to the signal. If we sum the intensity contributed by all directions, the points with the maximum intensities are the points of the resonance sources.

Fourier transform method.

The Fourier transform method [Kumar et al., 1975, Smith, 1985] is the widely used method in MRI. Instead of frequency, the location of the resonance source is encoded into the phase of the signal. We do not describe this in detail.

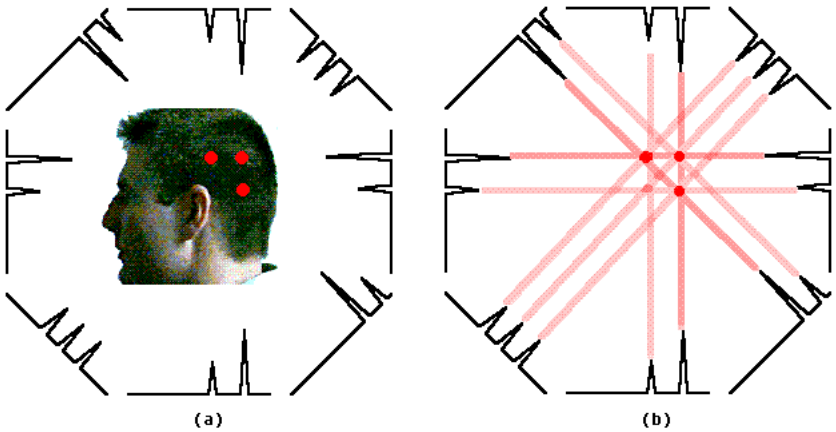


Figure 2.6: From [Hornak, 2005]. (a) 3 resonance sources in brain. Frequency encoding is conducted at different angles. (b) The signals of frequency encoding are projected back in space, and the points where all the projection lines intersect are located as resonance sources.

2.1.3 fMRI

MRI technology was designed as an imaging technique to detect the physical structures of subjects. However, since the work described in [Kwong et al., 1992, Ogawa et al., 1992], the MRI has also been used to investigate functionality of brain regions.

Principle.

One of the first hypotheses about "thinking" is that more blood flow would be observed during thinking. An early informal observation can be found in the English novel: "A Tale of Two cities".

> ... Both resorted to the drinking-table without stint, but each in a different way; the lion for the most part reclining with his hands in his waistband, looking at the fire, or occasionally flirting with some lighter document; the jackal, with knitted brows and intent face, so deep in his task, that his eyes did not even follow the hand he stretched out for his glass- which often groped about, for a minute or more, before it found the glass for his lips. Two or three times, the matter in hand became so knotty, that the jackal found it imperative on him to get up, and steep his towels anew. From these pilgrimages to the jug and basin, he returned with such eccentricities of damp headgear as no words can describe; which were made

> the more ludicrous by his anxious gravity ... – Charles Dickens, "A Tale of Two Cities" Dickens [1859]

An experiment was designed as early as 1890 [James, 1890] to test this hypothesis.

Neural activities increase the blood flow. More blood flow brings more oxygen, and the deoxy-Hemoglobin level declines, which causes the related signals change which can be detected by MRI devices. This type of fMRI, so-called "Blood Oxygen Level Dependent" fMRI, is the most popular fMRI technology.

The second key assumption is that the different brain parts are related to different cognitive tasks. So when we conduct different experiments, we expect different brain parts to be activated.

Unlike anatomical MRI, an fMRI experiment produces a sequence of brain images. Experiments usually last several minutes, and a 3D image is built every few seconds. To allow this fast scan, fMRI uses a special MRI technology called "EPI" (Echo Planar Imaging) [Forzaneh et al., 1990]. The fMRI resolution is lower compared to anatomical MRI. A example fMRI image is shown in Figure 2.7.

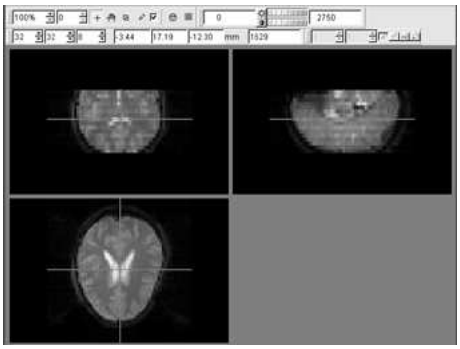


Figure 2.7: A sample fMRI images shown in FSL 3.2.

Experimental Design.

An fMRI experiment is designed to distinguish brain regions which are more sensitive to certain conditions. The major content of the experimental design is how to control the timing and duration of the stimulus, so that the "detection rate" can be maximized. Figure 2.8 shows a example of fMRI experimental design. The two types of stimulus and the "rest" condition are applied alternately.

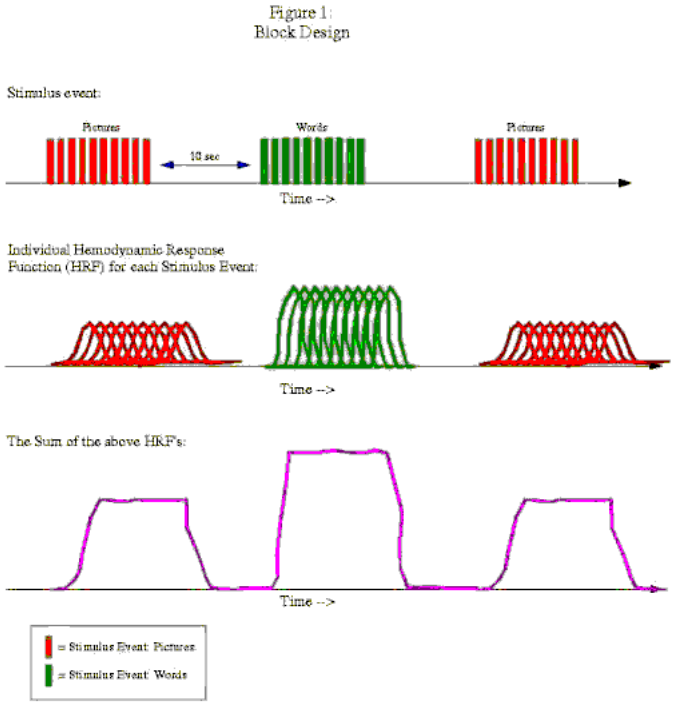


Figure 2.8: fMRI experimental design (Block design)

There are two types of fMRI experiments. The first is called "block design" (in figure 2.8). The condition is turned on-off alternately in time blocks. This on-off pattern is applied in whole experiment processes to cancel out noise. There are two limitations to this model. First, the subject may not have the same response in the middle of the experiment as that at the beginning of the experiment because he/she may get "used to" it. Second, the subject may discover the pattern of the stimulus, and grow an "expectation" of the next stimulus, which will invalidate

the timing assumption of the stimulus and the response. The second type of experimental design, which is called "event-related" design, can address these drawbacks. In event-related design, discrete stimulus will appear in arbitrary positions. An example can be seen in Figure 2.9. However, the event-related design usually causes a smaller signal noise ratio, and makes the data analysis more difficult.

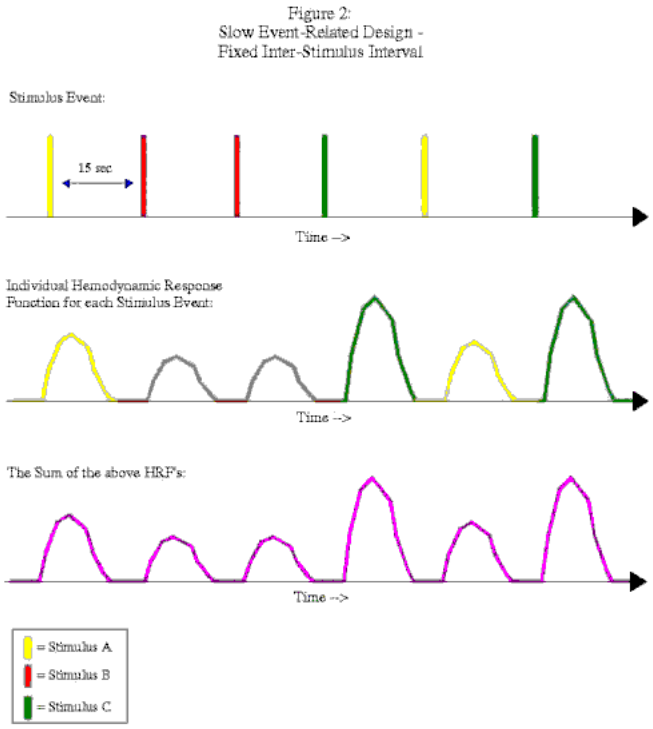


Figure 2.9: fMRI experimental design (Event-related design)

Chapter 3
Analysis of fMRI data

We can imagine that when a cognitive process occurs in the brain, the intensity somewhere in this image will change due to blood flow behavior over time. The intuition is to study the image in voxels. If a voxel is in a functional area for an invoked process, the time series of this voxel should have some kind of relationship with stimuli.

3.1 Preprocessing

The raw fMRI images as collected are not yet ready for analysis. Since the analysis methods so far are mostly based on time series for voxels, the consistency of brain volume position over time is very important. For example, if the head of the subject moves during data collection, the time series of a voxel may actually consist of several segments from different "real" voxels. Unfortunately, despite efforts at head fixation, there will always be head motion in fMRI experiments. The preprocessing to eliminate such motion effects is called *motion correction*.

Another problem is that the shapes of brains are different from person to person. For inter-subject functional studies, we need to have a map from each individual brain space to a standard space such as the MNI template. This procedure is called *registration*.

Motion correction and *registration* use similar technologies. Most methods define a "cost function" which describes the displacement between the experimental images and standard templates. The variables in cost function can be chosen to

be the affine transformations (translation, rotation, scaling, shearing). Then one uses optimization methods such as the gradient descent method to minimize the function value by adjusting the parameters.

A detailed review can be found in [Maintz and Viergever, 1998].

3.2 Hypothesis Driven (General Linear Model)

3.2.1 Hemodynamic Response Function

Here we discuss the problem at the voxel level only. A brain voxel is assumed to be an LTI (Linear Time Invariant) system [Oppenheim et al., 1983] subject to noise. The input of the system is the stimulus, the response of the system is this stimulus convolved with an impulse response function [Oppenheim et al., 1983]. The impulse response function is called the HRF (Hemodynamic Response Function) for this voxel.

The nature of the HRF has been extensively studied [Frackowiak et al., 2004]. A widely used assumption for the HRF is that its shape should be somewhat as in Figure 3.1(a), which is called the "canonical HRF" [Frackowiak et al., 2004]. Specifically, when a cognitive event occurs, initial consumption of oxygen causes a small dip. The level of refreshing blood oxygen reaches a peak in about 4-6 seconds, and then falls back to "idle" mode (with an apparent undershoot) gradually after that. The canonical HRF shown is generated as the difference of two gamma probability density functions:

$$H(t) = f(t; 6, 1) - \frac{1}{6} f(t; 16, 1), \tag{3.1}$$

where $f(t; \alpha, \beta) = \frac{1}{\beta^\alpha \Gamma(\alpha)} t^{\alpha-1} e^{-t/\beta}$ for $t > 0$. This model is often referred to as the "double gamma function". The first term controls the major shape of the "hill", while the second one is responsible for the undershoot. Figure 3.1(b), a "single gamma function", is a simplified version with the second gamma function

removed. Both models are provided in fMRI processing packages such as FSL [Smith et al., 2001].

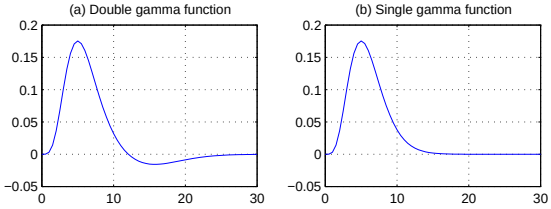


Figure 3.1: Canonical Hemodynamic Response Function.

The HRF is the foundation for hypothesis-based methods such as GLM (see 3.2.2). Thus it is still an important topic in fMRI study. Finite Impulse Response (FIR) models were suggested to approximate real HRFs [Woolrich et al., 2004, Goutte et al., 2000].

3.2.2 GLM

The General Linear Model (GLM) [Fox, 1984] is the most studied and most widely used method in the fMRI area. [Friston et al., 1994] and the related software package SPM (Statistical Parametric Maps) has had a great influence in fMRI study. Almost all the mainstream fMRI analysis software such as "AFNI" (Analysis of Functional NeuroImages) [Cox, 1996] and "FSL" [Smith et al., 2001] support GLM analysis.

In the GLM, observations (the time dependence of signal at each voxel) $\mathbf{y}$ are to be explained by the time courses of the $\mathbf{X}$ in form

$$\mathbf{y} = \mathbf{X}\mathbf{b} + \epsilon. \tag{3.2}$$

$\mathbf{X}$ is called the design matrix, in which every column is an "Explanatory Variable"(EV) generated by convolving the HRF with a "condition stimulus" time series. Some of the EVs are correlated to the experiment stimuli, for example,

the convolution of HRF with stimuli. Some of them are non-interesting components such as modeled noises.

The weight vector for stimuli in the observations $\mathbf{b}$ has the mean estimate $\hat{\mathbf{b}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$, and variance estimate $\hat{\epsilon}^2_{\mathbf{b}} = \epsilon^2(\mathbf{X}'\mathbf{X})^{-1}$, where $\epsilon^2 = \frac{RSS}{n-k-1}$. $RSS = (\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}})$ represents the Residual Sum of Squares, n is the number of observations (i.e. the length of $\mathbf{y}$), and k is the rank of $\mathbf{X}$. To determine if an element of $\mathbf{b}$ is significantly different from 0, we can perform statistical test. Specifically, for each voxel, a t-value can be calculated by dividing an element in $\mathbf{b}$ by its standard deviation, an element of $\epsilon_{\mathbf{b}}$, to indicate the significance of the voxel's activation by the corresponding condition. A 3D image of these t-values for all brain voxels is referred to as a "t-map". A large t value indicates more significant difference from 0. By setting threshold on t, we can select voxels above certain threshold level.

3.2.3 Finite Impulse Response model

While the canonical HRF works very well in many experiments, it is believed that the real HRF varies in different people, and in different regions of a person's brain [Frackowiak et al., 2004]. More flexible models, such as the Finite Impulse Response (FIR) model, have been proposed [Goutte et al., 2000]. In this model, the activation of a certain voxel at time t is the weighted sum of the stimuli $(s_i, i \in [t-n+1, t])$ at the preceding n time points. Formally,

$$y_t = \sum_{i=1}^{n} w_i s_{t-(i-1)} + w_0 + \epsilon, \tag{3.3}$$

where y_t is the intensity value at time t, ϵ is guassian noise, and w_0 is the constant component. The optimal estimate of $\mathbf{w} = [w_0, w_1, w_2, ...w_n]^T$ is defined as the one that minimizes the squared error between the observations and the model:

$$\mathbf{w}_{opt} = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_{t=n}^{N} (y_t - \hat{y}_t(\mathbf{w}))^2, \tag{3.4}$$

where

$$\hat{y}_t(\mathbf{w}) = \sum_{i=1}^{n} w_i s_{t-(i-1)} + w_0, \tag{3.5}$$

and N is the length of observation/stimuli sequence. This is a linear regression problem. Written in matrix form, Eq. (3.3) becomes: $\mathbf{Sw} = \mathbf{y} + \epsilon$, where

$$\mathbf{S} = \begin{pmatrix} 1 & s_n & s_{n-1} & ... & s_1 \\ 1 & s_{n+1} & s_n & ... & s_2 \\ 1 & ... & ... & ... & ... \\ 1 & s_N & s_{N-1} & ... & s_{N-n+1} \end{pmatrix},$$

$\mathbf{w} = [w_0, w_1, w_2, ..., w_n]^T$, and $\mathbf{y} = [y_n, y_{n+2}, ..., y_N]^T$, and the optimal estimate of $\mathbf{w}$ is $\mathbf{w} = [\mathbf{S}^T\mathbf{S}]^{-1}\mathbf{S}^T\mathbf{y}$. Compared to the canonical HRF, whose shape is basically fixed, this "naive" FIR model is very flexible, and thus has the potential to describe a greater variety of hemodynamic behaviors. However, this model tends to overfit as the number of parameters increases, especially when there is considerable noise, which is exactly the case for fMRI data. A typical observation is that large positive and large negative weights may appear alternately, as we will show in section 11.1. To overcome this problem, Goutte et al. [Goutte et al., 2000] proposed using a prior distribution $f(\mathbf{w}) \sim N(0, \Sigma)$, and finding maximum *a posteriori* (MAP) parameter estimation:

$$\mathbf{w}_{MAP} = (\mathbf{S}^T\mathbf{S} + \sigma^2\Sigma^{-1})^{-1}\mathbf{S}^T\mathbf{y}, \tag{3.6}$$

where $\Sigma_{ij} = v \exp(-\frac{h}{2}(i-j)^2)$, in which h is the smoothing factor, v is the strength of smoothing, σ^2 is the variance of the noise. See appendix B for a detailed derivation of eq 3.6. The correlation imposed among parameters prevents sudden changes (spikes) in the form of the HRF.

Although the "smoothing" in this Bayesian FIR model can eliminate drastic changes to some degree, it has some limitations: First, it requires careful selection of the hyper parameters, h, v and σ^2. Second, and more important, although MAP

estimation can smooth changes between neighboring parameters, our experiments find that it can not remove large negative values and multiple peaks without making the whole HRF too flat.

We propose to address these problems by injecting prior knowledge about the shape of the canonical HRF into FIR models, with some (desirable) loss of flexibility. Specifically, we impose shape information derived from the single gamma HRF on the FIR model, as a set of linear constraints. Thus, the linear/Bayesian regression problem becomes a constrained quadratic optimization problem.

3.3 Data Driven (Principal Component Analysis and Independent Component Analysis)

3.3.1 Principal Component Analysis (PCA)

PCA [O'Connell, 1974, Bishop, 1995] is an algorithm to find directions with largest variances. Give a set x of l vectors, the first PCA component can be computed with equation 3.7.

$$w_1 = \arg\max_{\|w\|=1} E\{(w^T x)^2\} \tag{3.7}$$

In Figure 3.2 we can see the direction of w_1 is aligned at the direction on which the projection of the data has the largest variance.

The next principal component will be calculated in the same process with the previous principal component removed from dataset. Suppose we already have w_1 to w_{k-1}, then

$$w_k = \arg\max_{\|w\|=1} E\{[w^T(x - \sum_{i=1}^{k-1} w_i w_i^T)x]^2\} \tag{3.8}$$

PCA is usually used to reduce the dimension of data. Each data point in n- dimension space R^n, it has n coordinates in n axes. However, some of these coordinates are strongly correlated, this is considered as data redundancy. Under

Figure 3.2: From [Hyvarinen, 1999]. The first principal component of a 2D dataset.

certain circumstances, the data can be transformed into a lower dimension space, by dropping some directions with small variances. The coordinates in the new system are the projections of the old coordinates on the principal directions.

To find the principal components for n-dimensional data, the usual approach is to build the $n \times n$ covariance matrix C for coordinates of the dataset first, and then eigenvalues and eigenvectors are calculated for C. If we want k principal components, we take the eigenvectors corresponding to first k largest eigenvalues.

The coordinates respective to principal component are not correlated. That is, for data points with component coordinates

$$C_i = c_i, i = 1, \cdots, n, E[(C_i - \overline{C_i})(C_j - \overline{C_j})] = 0, i \neq j$$

. This is also referred to as "second order independence".

PCA can be used in different ways in fMRI study. In Le and Hu [1995], Mitra et al. [1995], PCA is used to detect latent spatial activation patterns in the brain by find principal components of matrix A where each element $A_{i,j}$ is the intensity of voxel j in volume i in the same fMRI scan. In [Ford et al., 2003], PCA is used on the activation maps built from different experiments to detect patterns across experiments and subjects. More details about this use are presented in chapter

8.

3.3.2 Independent Component Analysis (ICA)

ICA [Hyvarinen, 1999, Hyvarinen and Oja, 2000] is a different method to select important latent components behind the data. Independent components requires statistically independence, which is defined as:

$$p(C_1 = c_1, C_2 = c_2, \cdots, C_n = c_n) = \prod_{i=1}^{n} p(C_i) = c_i \tag{3.9}$$

. An equivalent definition is:

$$E\{g_1(C_1)g_2(C_2)\} = E\{g_1(C_1)\}E\{g_2(C_2)\} \tag{3.10}$$

for any functions g_1 and g_2.

To see the difference, consider $C_1 = C_2^2$, the correlation between C_1 and C_2 is 0, but they are clearly not independent.

Formally, the problem of ICA is presented in equation 3.11 or 3.12, depending on whether the noise is considered as one independent component.

$$\mathbf{Y} = \mathbf{BX} + \eta \tag{3.11}$$

$$\mathbf{Y} = \mathbf{BX} \tag{3.12}$$

Rows of $\mathbf{Y}$ are observation vectors. Rows in $\mathbf{X}$ are latent components. In ICA, these components are independent. $\mathbf{B}$ are the mixture weights of components in $\mathbf{Y}$. η is the noise. This equation is similar to Eq3.2. First, they are both in linear form. Second, the matrix ($\mathbf{X}$ in equation 3.2 and in 3.12) contains the components, except that these components are columns in Eq. 3.2, but are in rows in Eq. 3.12. We write ICA formulation in this way mostly because we want to be consistent with the later spatial ICA model presented in 3.3.2.

The difference is, unlike GLM, in which $\mathbf{X}$ is the design matrix prepared by researchers, ICA does not know either parameters $\mathbf{B}$ or components $\mathbf{S}$. $\mathbf{B}$ and

S have to be calculated together. Thus, ICA is very closely related to an earlier research topic "Blind Source Separation" [Jutten and Herault, 1991].

Most ICA research considers the noiseless model. What ICA does is to find a matrix **W** such that we can get **U** from the linear transformation 3.13:

$$\mathbf{U} = \mathbf{WY} = \mathbf{WBX} \tag{3.13}$$

such that **U** is an approximation of **X**, assuming that columns in **X** really are independent. Ideally, WA is in the form of $\mathbf{P} \cdot \mathbf{X}$, in which **P** is a permutation matrix, while **X** is a diagonal matrix representing the scaling. In other words, **U** is exactly **X** except that the magnitude and orders are changed.

If we know **B** is a invertible matrix, then **W** is the inverse of **B**. However, **B** is unknown to us, so we have to find approximate solutions.

The idea is, we want to find out **W** to make rows in **U** as statistically independent as possible. However, the definition of statistical independence 3.9 does not help because it is hard to estimate probability density function from discrete sample data. Instead, we need to find alternative indicators of independence. Moreover, we need algorithms to find **W** to maximize this indicator. Thus, ICA consist of two critical parts: the objective functions and the optimization algorithms.

Objective Function

There are two basically categories of objectives: mutual information based on information theory, and non-Gaussianity based on central limit theorem.

We define random variable $\mathbf{u} = (u_1, u_2, ..., u_m)$ such that an element (i, j) of **U** is the jth observation of the random variable u_i. Mutual information [Mackay, 2003] defined in information theory is a natural choice for independence:

$$I(\mathbf{u}) = I(u_1, u_2, \cdots, u_m) = \sum_{i=1}^{m} H(u_i) - H(u_1, u_2, \cdots, u_m) \tag{3.14}$$

Here $H(\mathbf{u})$ is the differential entropy

$$H(\mathbf{u}) = \int_{\mathbf{u}} p(\mathbf{u}) \ln p(\mathbf{u}) d\mathbf{u}$$

in which $p(u)$ is probability density function. The mutual information $I(\mathbf{u})$ is always greater than 0, and it reaches 0 only when $u_i, (i = 1, 2, \cdots, m)$ are independent.

Non-Gaussianity is another important indicator of independence. Although we could not find any mathematical proof, the following assumption is used widely in ICA study.

> The central limit theorem ... says that the distribution of a sum of independent random variables tends toward a Gaussian distribution, under certain conditions. Loosely speaking, a sum of two independent random variables usually has a distribution that is closer to Gaussian than any of the two original random variables.
> –Hyvarinen et al. 2001 [Hyvarinen et al., 2001]

Kurtosis is a classical metric for non-Gaussianity (see equation 3.15).

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{3.15}$$

In practice, the random variable y is usually normalized with variance 1, thus equation 3.15 is simplified to

$$kurt(y) = E\{y^4\} - 3 \tag{3.16}$$

Kurtosis is 0 for Gaussian distribution. If kurtosis is greater than 0, the distribution is called "super Gaussian", and is considered more "spiky" than Gaussian (see Figure 3.3a). If kurtosis is less than 0, the distribution is called "sub Gaussian", and is more "flat" than Gaussian (see 3.3b).

Negentropy is another non-Gaussian metric, which is defined as:

$$J(y) = H(y_{gauss}) - H(y) \tag{3.17}$$

y_{gauss} is the Gaussian distribution which has the same variance as y. Negentropy is a nonnegative distance from Gaussianity because Gaussian distribution has the

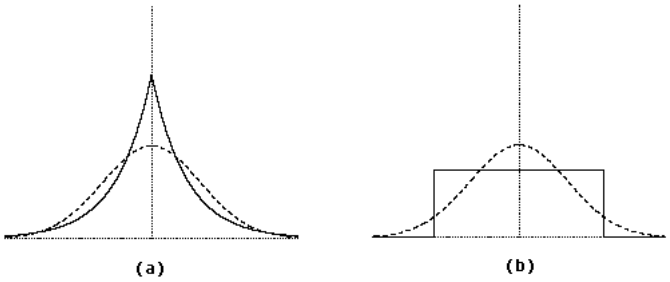


Figure 3.3: From [Hyvarinen et al., 2001]. (a) A super Gaussian distribution. (b) A sub Gaussian distribution

largest entropy among all distributions given the variance. Estimating entropy from sample data is not preferred, an approximation was described in [Hyvarinen et al., 2001].

Optimization algorithms

There are many algorithms proposed for ICA. A good review can be found in [Hyvarinen and Oja, 2000]. Here we introduce INFOMAX approach proposed in [Bell and Sejnowski, 1995], which is also the earliest ICA algorithm applied in fMRI research [McKeown et al., 1998], and FastICA [Hyvarinen et al., 2001], based on fixed point iteration method on kurtosis negentropy.

INFOMAX method

Direct optimization of Equation 3.14 is not appropriate since the joint distribution of $u_i, i = 1, 2, \cdots, m$ is not easy to get. Instead, [Bell and Sejnowski, 1995] showed that minimization of $I(\mathbf{u})$ can be replaced by maximizing the joint entropy of output of sigmoidal nonlinear function $\mathbf{y} = g(\mathbf{u}) = (g_1(u_1), g_2(u_2)..., g_n(u_n))$, based on INFOMAX principle [Linsker, 1988]. There are choices for sigmoid function $g_i(u_i), i \in [1, m]$, the popular ones are logistic function

$$g_i(u_i) = \frac{1}{1 + e^{-u_i}} \tag{3.18}$$

and hyperbolic tangent function:

$$g_i(u_i) = tanh(u_i) \tag{3.19}$$

First Let's look at a simple case that both x and y are scalar variables, and $u = wx + w_0$). When $g(u)$ is a monotonically increasing or decreasing function, the p.d.f. $f_y(y)$ is written as:

$$f_y(y) = \frac{f_x(x)}{|\partial y/\partial x|} \tag{3.20}$$

And entropy of y is:

$$H(y) = -E\left[\ln f_y(y)\right] = -\int_{-\infty}^{\infty} f_y(y) \ln f_y(y) dy \tag{3.21}$$

By plugging 3.20 into 3.21, we have

$$H(y) = E\left[\ln\left|\frac{\partial y}{\partial x}\right|\right] - E\left[\ln f_x(x)\right] \tag{3.22}$$

We want to use gradient descent method to find optimal w. In other words, we interactively change w in direction of decreasing $H(y)$, that is

$$\Delta w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}(\ln\left|\frac{\partial y}{\partial x}\right|) = (\frac{\partial y}{\partial x})^{-1}\frac{\partial}{\partial w}\frac{\partial y}{\partial x} \tag{3.23}$$

We noticed that in this derivation, the expectation $E\left[\ln\left|\frac{\partial y}{\partial x}\right|\right]$ is replaced by $\ln\left|\frac{\partial y}{\partial x}\right|$. In other words, the expectation is replaced with single sample points. This is referred to as stochastic gradient method [Hyvarinen et al., 2001]. It has been argued that the algorithm converges to the same point, if the updating step is chosen to be small enough and a lot of samples are used.

The second term in equation 3.22 is ignored because it does not change with different w. When we use logistic function 3.18, we have

$$\Delta w \propto \frac{1}{w} + x(1 - 2y) \tag{3.24}$$

and

$$\Delta w_0 \propto 1 - 2y \tag{3.25}$$

Now, suppose $\mathbf{x}$ and $\mathbf{y}$ are size $\mathbf{n}$ vectors, and $\mathbf{y} = g(\mathbf{u}), \mathbf{u} = \mathbf{W}\mathbf{x} + \mathbf{w}_0$, by similar deduction which is described in [Bell and Sejnowski, 1995], we have

$$\Delta\mathbf{W} \propto \left[\mathbf{W}^T\right]^{-1} + (\mathbf{1} - 2\mathbf{y})\mathbf{x}^T \tag{3.26}$$

$$\Delta\mathbf{w}_0 \propto \mathbf{1} - 2\mathbf{y} \tag{3.27}$$

In 3.26 and 3.27, $\mathbf{1}$ is a vector with all one's.

FastICA

FastICA takes non-Gaussianity metrics such as kurtosis or negentropy as the indicator of independence. The negentropy approach can be seen as a generalization of kurtosis approach, and have the merit of better performance for outliers. To the principle with simplicity, we introduce the kurtosis approach here.

The kernel of FastICA is to minimize the Gaussianity of a single component. In other words, it tries to find a direction $\mathbf{w}$ in which the projection of data samples $\mathbf{x}$ has the the minimum Gaussianity. That is equivalent to finding the maximum absolute value of kurtosis $\left|kurt(\mathbf{w}^T\mathbf{x})\right|$. So either the components can be found one by one, in a way similar to PCA, or some symmetric method (which we do not discuss here) can be applied to approximate all components at the same time. To regularize later computation, we apply a constraint:

$$\|\mathbf{w}\| = 1 \tag{3.28}$$

Before ICA is applied, two preprocessing are usually applied. The first is decorrelation, often called "whitening", this step can be done with PCA. The second is called "sphering" which is to make variance in each axis direction to be 1. These two steps can be written together in a linear transformation $\mathbf{V}$. In following discussions, we consider $\mathbf{z} = \mathbf{V}\mathbf{x}$ as the input to ICA algorithm. We will see the convenience of this preprocessing.

Kurtosis is defined in equation 3.15:

$$kurt(\mathbf{u}) = E\{\mathbf{u}^4\} - 3(E\{\mathbf{u}^2\})^2$$

, while $\mathbf{u} = (\mathbf{w}^T\mathbf{z})$. When reaching maximum we have:

$$\frac{\partial\,|kurt(\mathbf{w}_T\mathbf{z}|}{\partial\mathbf{w}} = 4sign(kurt(\mathbf{w}^T\mathbf{z}))[E\{\mathbf{X}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}E\{(\mathbf{w}^T\mathbf{z})^2]\} \quad (3.29)$$

$$= 4sign(kurt(\mathbf{w}^T\mathbf{z}))[E\{\mathbf{X}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}\,\|\mathbf{w}\|^2] \quad (3.30)$$

The second equation was due to the fact that $\mathbf{z}$ is sphered.

To apply fixed point iteration, we need to find a representation of $\mathbf{w}$ as a output of its own function:

$$\mathbf{w} = f(\mathbf{w}) \quad (3.31)$$

An interesting fact is that $f(\mathbf{w})$ should be $\frac{\partial\left|kurt(\mathbf{w}^T\mathbf{z})\right|}{\partial\mathbf{w}}$. This is because under the constraint $\|\mathbf{w}\| = 1$, the minimization of $\frac{\partial\left|kurt(\mathbf{w}^T\mathbf{z}\right|}{\partial\mathbf{w}}$ should satisfy the following equation:

$$\frac{\partial\left|kurt(\mathbf{w}^T\mathbf{z})\right|}{\partial\mathbf{w}} - \lambda\frac{\partial}{\partial\mathbf{w}}(<\mathbf{w},\mathbf{w}> -1) = 0 \quad \Rightarrow \quad (3.32)$$

$$\frac{\partial\left|kurt(\mathbf{w}^T\mathbf{z})\right|}{\partial\mathbf{w}} - 2\mathbf{w} = 0 \quad (3.33)$$

So we know at the stationary point, $\mathbf{w}$ and $\frac{\partial\left|kurt(\mathbf{w}^T\mathbf{z})\right|}{\partial\mathbf{w}}$ are at the same direction. Since we normalize $\mathbf{w}$ in every iteration, and only the direction is interesting, we can get iteration rule:

$$\mathbf{w} = 4sign(kurt(\mathbf{w}^T\mathbf{z}))[E\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}\,\|\mathbf{w}\|^2] \quad (3.34)$$

$$= 4sign(kurt(\mathbf{w}^T\mathbf{z}))[E\{\mathbf{z}(\mathbf{w}^T\mathbf{z})^3\} - 3\mathbf{w}] \quad (3.35)$$

Figure 3.4 shows an example of how FastICA separates the sources in 4 steps.

ICA Application in fMRI

ICA was introduced to the fMRI community by [McKeown et al., 1998]. Considerable work has been done since, please see [Calhoun et al., 2003] for review.

As mentioned earlier, ICA is similar to GLM in terms of its linear form. In Figure 3.5, we show the connection between these two models. Figure 3.5 (a) is GLM with one voxel. The time series of a voxel is the linear combination of the vectors in the known design matrix. Figure 3.5 (b) is the GLM representation of all the voxels put together. Note that the design matrix $\mathbf{X}$ is the same. Figure 3.5 (c) is like (b), except that the design matrix $\mathbf{X}$ is not known any more. Blind separation needs to be performed.

We have two options here. First, we can adjust $\mathbf{B}$ to make the columns in $\mathbf{X}$ independent. Since the columns have a length same as the voxel time series, they can be considered as independent time courses, representing the different wave forms of brain processes. This method is called temporal ICA. Second, we can adjust $\mathbf{X}$ to make rows in $\mathbf{B}$ independent. Although it is as intuitive, we can imagine that the responses of brain regions to independent brain processes should be independent. Since each row of $\mathbf{B}$ has all the voxels, it can be considered as some activation map, corresponding to the time courses in the columns of $\mathbf{X}$. This method is called spatial ICA. Since a fMRI experiment has usually a few hundreds volumes, spatial ICA is preferred in most analysis since it provides more samples. However, some research [Calhoun et al., 2001] shows temporal and spatial ICA can produce comparable results. Another reason spatial ICA is preferred is that adjusting a smaller matrix $\mathbf{X}$ is much more efficient than a larger $\mathbf{X}$ in solution searching.

Probabilistic ICA (PICA) [Beckmann and Smith, 2004] proposed a "noisy" ICA model, in a way similar to the general linear model:

$$\mathbf{Y} = \mathbf{BX} + \eta$$

in which η is Gaussian noise. Instead of making $\mathbf{B}$ a full rank square matrix, the dimension is reduced first using probabilistic PCA (PPCA) [Tipping and Bishop, 1997]. Thus, $\mathbf{Y}$ is not fully explained as a linear combination of the rows in $\mathbf{X}$ as in the noiseless ICA model. The residual is considered as noise, and is used

in statistical inference. An ICA component includes a time course (which is a column in **B** and a brain map in which each voxel has a z-value that indicates the activation of the voxel. This statistical map is referred to as *z-map*. PICA is implemented in fMRI analytical software FSL as the "melodic" module, and it is the ICA model we use here.

3.4 Classification

In recent years, machine learning methods in fMRI analysis have attracted attention. [Mitchell et al., 2004, LaConte et al., 2005, Ford et al., 2003] tried machine learning approaches on fMRI data for different purposes.

One of the major difficulties of machine learning methods in fMRI research is feature representation. Usually an fMRI dataset contains a few hundred 3D brain scans, and each scan contains more than 10,000 voxels. Meanwhile, the number of datasets is typically less than 100. If the voxel values are used as features in a brute force way, it will cause an over fitting problem. Thus, some of these studies used data reduction in some way.

[Mitchell et al., 2004] tried to use machine learning method to distinguish one condition from another. To reduce the data size, they performed training on each voxel as a classifier, and only selected the voxels which distinguish the conditions best as features. SVM, Naive Bayesian and kNN methods are evaluated in their experiments.

[Ford et al., 2003] tried to address almost the same problem as [Mitchell et al., 2004]. More specifically, they tried to distinguish datasets associated with different diseases. They first perform SPM to build a t-map (see 3.2.2) for each condition, then this t-map is reduced with PCA. Fisher's linear discriminant method [Fisher, 1936] is trained as a classifier. Fisher's linear discriminant (FLD) projects high dimensional points in to a line. On this line, FLD tries to maximize

the distance of mean values between two classes, while minimizing the variance inside each class.

[Ford et al., 2003] worked on a different task. They tried to separate brain volumes under different conditions in the same time series. Another key difference from [Ford et al., 2003] and [Mitchell et al., 2004] is that int this work voxels are selected based on the dataset alone, and not on their ability to distinguish among conditions that happen to be in the training set.

Note that all the above methods are voxel-based, so the performance of the methods are affected by motion correction and registration.

3.5 Wavelet-based analysis

It is well known that the Fourier transform decomposes a time and/or space signal into sinusoidal signals with different frequencies. This can also be considered as a linear transform with orthogonal sinusoidal bases. Similarly, the wavelet transform [Mallat, 1989] is a linear transform with orthogonal bases. Unlike Fourier bases, the wavelet bases do not span the whole domain. Instead, a wavelet basis has non-zero values only on a bounded region. So, each component of a basis catches only signal in a that area. We shall see that this is a major advantage of wavelet transform over Fourier transform when handling transient signals.

Any wavelet basis is built based on basic forms called the "mother" function or wavelet function $\psi(x)$, and a scaling function or "father" function $\phi(x)$. By dilating and translating these wavelets one builds the whole set of wavelet bases.

There have been many types of wavelets proposed, Here we only show the simples type called Haar bases. The scaling and wavelet functions of Haar transform are:

$$\phi(x) = \begin{cases} 1 & : \quad 0 < x < 1 \\ 0 & : \quad otherwise \end{cases} \tag{3.36}$$

$$\phi(x) = \begin{cases} 1 & : \quad 0 < x < \frac{1}{2} \\ -1 & : \quad \frac{1}{2} < x < 0 \\ 0 & : \quad otherwise \end{cases} \tag{3.37}$$

The wavelet function is always scaled by a factor of 2, and translated to fill the domain. Figure 3.6 shows a example of a set of Haar bases.

There are different ways to think of wavelet transform. One intuitive understanding is that wavelets decompose the signals into coarse and detailed components recursively. Figure 3.7 shows an example of how a wavelet transform is performed. The original signal can be recovered by reversing this process.

As we mentioned before, fMRI has a variety of noise sources with different patterns, so the noise can not be eliminated by applying a simple low pass or high pass filter. Also, the signal is only a delayed response to the stimulus. In this case, wavelets are very attractive because of their multi-resolution analysis ability. [Bullmore et al., 2003] gives a good review of current application of wavelet to the fMRI area.

3.6 Notes

This chapter includes parts of [Bai and Kantor, 2007, Bai et al., 2007b,c], which are co-authored by Bing Bai, Paul Kantor, Ali Shokoufandeh, and Deborah Silver.

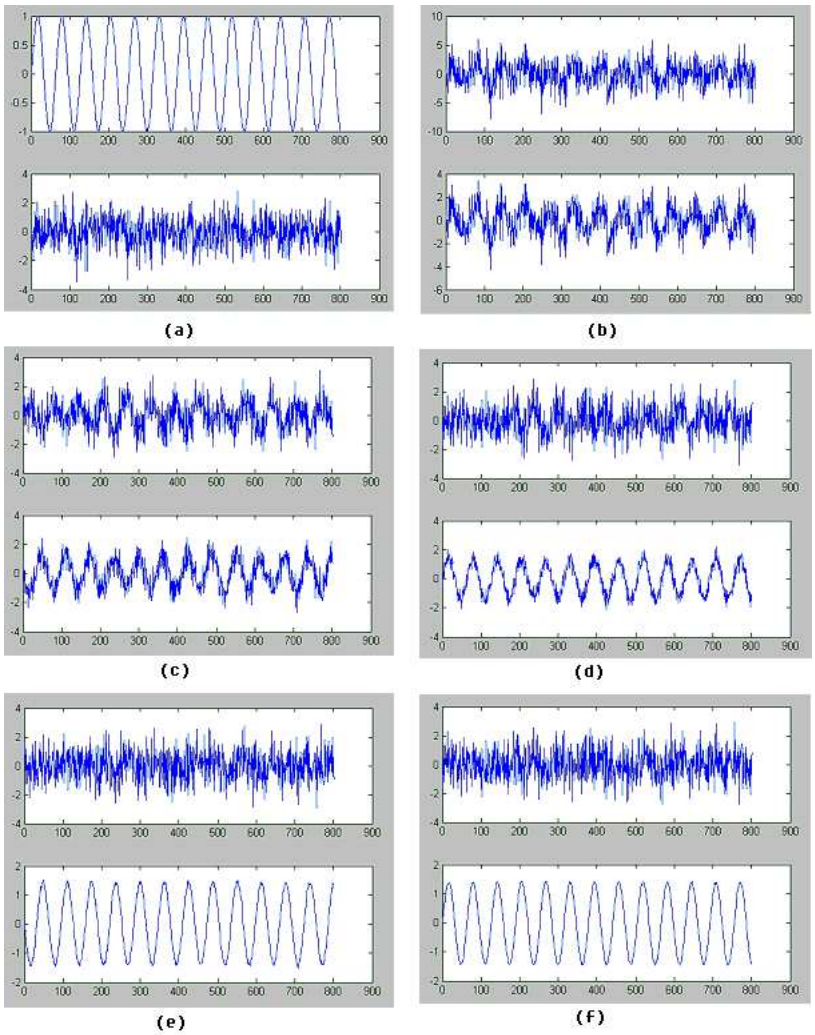


Figure 3.4: Source separation computed with FastICA [Hyvarinen, 2005]. (a) Original sources, a sine curve and a Gaussian noise. (b) 2 linear mixture of the sources. (c),(d),(d) and (f) are 4 iteration steps. The sum of the corresponding kurtoses of the two components are: (c) -0.9246, (d) -1.2858, (c) -1.5959, (d) -1.6081. We can see by maximizing the absolute value of kurtosis, independence between components is achieved.

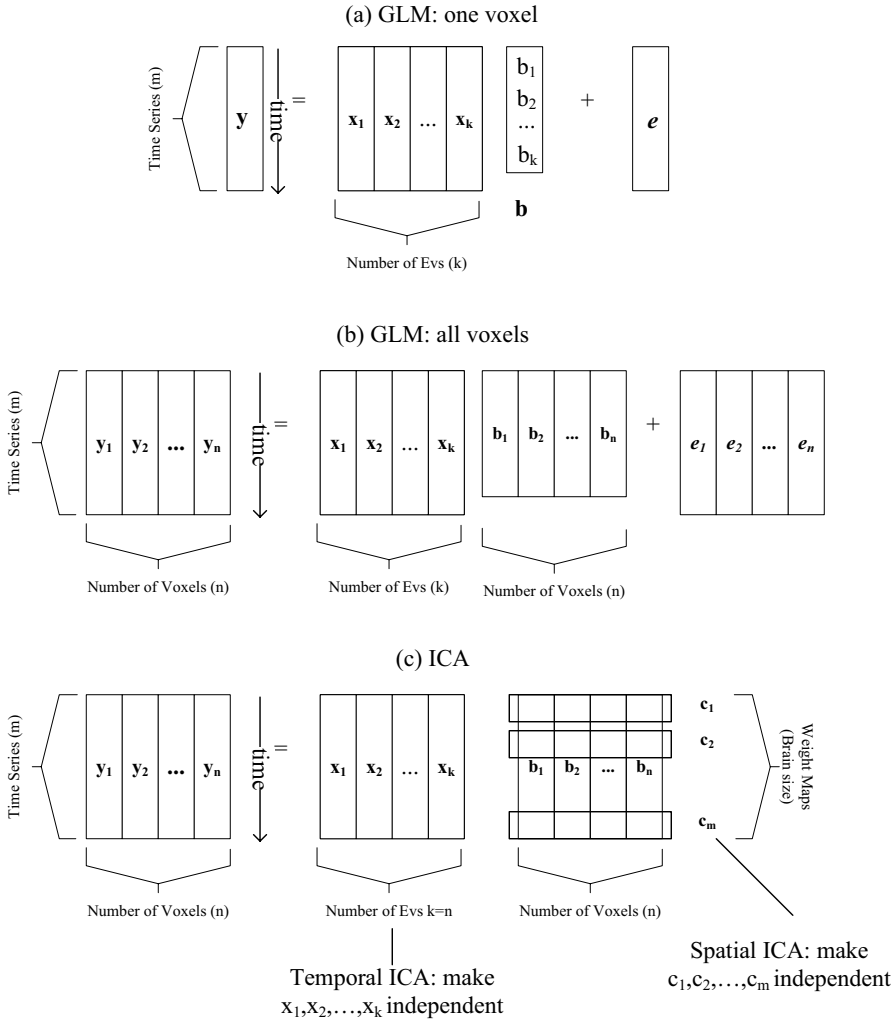


Figure 3.5: Comparison between GLM and ICA.

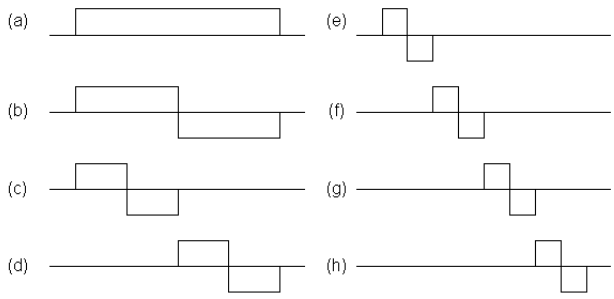


Figure 3.6: A set of Haar wavelet Bases

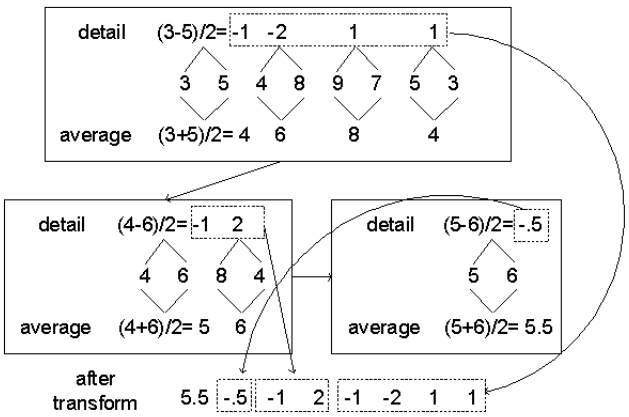


Figure 3.7: A example wavelet transform. Details and average are calculated for each adjacent pair, and the average is repeated recursively until only one element left. All differences, and the overall average make up the result.

Chapter 4
Feature Extraction Framework

Features are extracted from a dataset (such as a document or an image), to represent the whole dataset in further processing. A good feature extraction scheme should help in both retrieval effectiveness and efficiency. Note that retrieval effectiveness is not "discovery effectiveness", which suggests the connection between a brain region with a certain functionality. Retrieval effectiveness only indicates how much the selected features can improve the retrieval performance.

In "regular" image retrieval tasks, the most common features include color, shape, texture, etc. For fMRI, we need to find feasible features for us to index the images by experiment conditions, which represent underlying cognitive brain processes.

The features we seek, which will play the role of "terms" in retrieval should satisfy two criteria:

1. *Retrieval Effectiveness*: Selected regions should be more similar form experiments with same conditions.

2. *Efficiency*: Can be embedded into efficient algorithms.

In this thesis, the features we use can be described as "voxel-based" method. More specifically, we choose the features to be voxels that are most salient in responding to the stimulus. This method is based on two assumptions: 1. The functions of brain regions have some stability across individual human brains [Frackowiak et al., 2004]. 2. Only a small fraction of brain responds to simple brain processes. [Frackowiak et al., 2004].

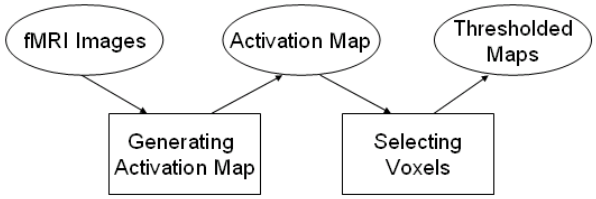


Figure 4.1: Feature selection is in two stages, the first stage is to build a brain map in which the value of each voxel indicates the activation, the second stage is to select most important of these voxels based on their activation levels.

We propose a feature selection method, as shown in Figure 4.1. This scheme can be viewed as a two-stage black box system.

The first stage is to build "activation maps", in which the value of each voxel indicates the level of activation of the voxel. The most common activation map is the t-maps from GLM (see 3.2.2). However, building activation maps is at the heart of much fMRI research and many methods are still under development. In fact, it is possible that the best methods for different experiments will be different.

The second stage selects the "most important" voxels. Intuitively, we should select voxels whose activation are "large enough", for voxels with low activation are more likely to be random than causally related to the stimulus. Of course, time and space cost increase as more voxels are included. There is no consensus on what t-value should be considered "large enough", so here we simply take that 1% of the voxels with largest t-values (or largest absolute t-values, depending on the method used, more details follow). Although this decision seems arbitrary, it has the virtue that the number of voxel representing different datasets tends to be nearly the same, which would not be true if a threshold were set based on the t-value itself. We call the image of the selected voxels the *thresholded t-map*.

Our retrieval model closely parallels classical text retrieval, with these correspondences:

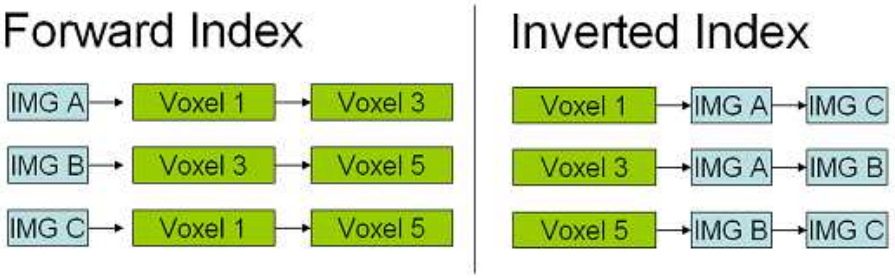


Figure 4.2: Forward and inverted index. Two voxels are activated in each brain images.

Brain images	*correspond to*	Documents
Activated voxels	*correspond to*	Terms
Query voxels	*correspond to*	Query Terms

With this similarity in mind, we can we build both forward and inverted indices. In the forward index, each entry is an image, with a link to its activated voxels. In the inverted index, each entry is a unique voxel (in the standard brain), pointing to a list of brain images in which that voxel is activated. This is shown in Figure 4.2.

In this chapter we mainly focus on the first stage, i.e. building activation maps. Up to now, canonical HRFs are still by far the most widely used hemodynamics model. However, there are several critical issues about GLM still under investigation. The two key factors for building explanatory variables are hemodynamic response function (HRF) and the stimuli. It is well known that HRF differs across subjects and brain regions, and it is hard to define the starting and ending time of the stimulus with a more complicated experimental design and more complex cognitive tasks. A false assumption about HRF or stimulus will directly affect the accuracy of activation detection, similarly with the stimulus. We propose to attack these problems at several levels.

1. We apply FIR model instead of fixed HRF. We propose constraints to make

the parameter estimation more robust. We also develop an algorithm to enable simultaneous estimation of the HRF and activation level for multiple conditions.

2. We drop the assumption of the LTI response model, and use the stimulus and a non-linear model based on an ordinary differential equation to estimate the explanatory variable directly. This non-linear model is somewhat similar to the balloon model [Buxton et al., 1998], but simplified to allow quick computation.

3. We drop the requirement of stimulus, as well as HRF, and discover the task related "explanatory variables" directly from ICA components. We propose several heuristics to identify the task related components.

4.1 Notes

This chapter includes parts of [Bai et al., 2006, 2007a], which are co-authored by Bing Bai, Paul Kantor, Nicu Cornea, and Deborah Silver.

Chapter 5
GLM-based New FIR Models

5.1 The Single Peak Non-Negative (SPNN) FIR Model

5.1.1 Model

The SPNN FIR model retains the objective function (3.4), and adds the following constraints: 1) there is a single peak at some delay, p, so that

$$\begin{aligned} w_{i-1} \leq w_i \quad &: \quad i \in [2, p] \\ w_i \geq w_{i+1} \quad &: \quad i \in [p, n-1] \end{aligned} \tag{5.1}$$

2) all weights are non-negative:

$$w_i \geq 0, i \in [1, n] \tag{5.2}$$

These two sets of constraints help us to keep the major shape features of the canonical HRF: 1) the majority of responses are positive; 2) the response rises to a single maximum and decays after that. Note that the initial dip and the recovery undershoot are lost in this model.

For each value of the location of peak p, the problem is then to minimize the objective function of (3.4) subject to the constraints (5.1) and (5.2). This is a quadratic programming problem. If we write the objective as follows:

$$\sum_{t=n}^{N} (y_t - \hat{y}_t(\mathbf{w}))^2 = \sum_{t=n}^{N} {y_t}^2 - \sum_{t=n}^{N} 2 y_t \hat{y}_t(\mathbf{w}) + \sum_{t=n}^{N} \hat{y}_t(\mathbf{w})^2,$$

we notice that the third term $\sum \hat{y}_t(\mathbf{w})^2$ contains all the quadratic terms and only the quadratic terms in the weights $w_i, i \in [0, n]$, and it is always non-negative.

This means that the problem is a semi-definite quadratic programming problem and a global optimal solution can be found via a variety of methods including the interior-point method and the active set method [Nocedal and Wright, 1999].

So far, we have assumed that the peak location p is given. In fact, to determine the location of the peak, we repeat the above procedure for each $p \in [1, n]$, and then retain the weights which generate the best fit.

SPNN and MAP are two approaches to dealing with the overfitting problem (SPNN uses shape information while MAP uses parameter correlation). It is also possible to combine them. To do this, we note that the MAP estimate (3.6) minimizes the following objective with no constraints:

$$\min \sum_{t=1}^{N} (y_t - \hat{y}_t(\mathbf{w}))^2 + \mathbf{w}^T \Sigma^{-1} \mathbf{w}. \tag{5.3}$$

The constant w_0 should not be penalized, thus the element (1,1) should be reset to 0 for Σ^{-1}. By imposing the linear constraints of (5.1) and (5.2) on the objective (5.3), we obtain a quadratic programming problem, whose solution may combine merits of SPNN and MAP estimation.

5.2 FIR model for multiple conditions at the same time

The aforementioned FIR model can only account for a single stimulus condition. However, it is quite common that several conditions are applied in a single fMRI run. Although we could deal with this by using each condition separately in simple regression, there is a potential problem with that approach.

Suppose several conditions have similar effects on one voxel. If we consider only one condition, then the residual sum of squares RSS will be greater than if we consider all conditions together, and result in a smaller t-value. In other words, voxels whose time series are in fact just noise have a better chance to be selected.

In order to address this problem, we propose to combine the FIR model with multiple regression. This will in turn allow us to simultaneously compute the estimates for the HRF and the activation levels. We assume that the shape of HRF is the same for different stimulus on a certain voxel, because HRF describes a physiological feature of a certain brain region, which should depend on how much the region is activated, not why it is activated.

5.2.1 Algorithm

Specifically, assume that we have c conditions, whose stimulus time series are: $s_j^i, i \in [1, c], j \in [1, N]$. Then an estimate for the activation can be written in the following parametric form: $y_t = \hat{y}_t + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$

$$\begin{aligned} \hat{y}_t &= \sum_{j=1}^{c} a_j \sum_{i=1}^{n} w_i s^j_{t-(i-1)} && (5.4) \\ &= \mathbf{a}^T \mathbf{S_t} \mathbf{w} && (5.5) \\ &= (a_1, a_2, ...a_c) \begin{pmatrix} s_t^1 & s_{t-1}^1 & ... & s_{t-(n-1)}^1 \\ s_t^2 & s_{t-1}^2 & ... & s_{t-(n-1)}^2 \\ ... & ... & ... & ... \\ s_t^c & s_{t-1}^c & ... & s_{t-(n-1)}^c \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ ... \\ w_n \end{pmatrix} && (5.6) \end{aligned}$$

The components of $\mathbf{a}$ are the weights of each condition, and $\mathbf{w}$ is the HRF. For clarity, we omit the constant term from Eq. 5.7.

We study the least squares fit of this bilinear model:

$$\mathbf{a}, \mathbf{w} = \underset{\mathbf{a},\mathbf{w}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \mathbf{a}^T \mathbf{A_i} \mathbf{w})^2 \tag{5.7}$$

Let $\mathbf{D}(\mathbf{a}, \mathbf{w}) = \sum_{i=1}^{n} (y_i - \mathbf{a}^T \mathbf{A_i} \mathbf{w})^2$. When $\mathbf{a}, \mathbf{w}$ reach a cricical point, we have

$$\begin{aligned} \tfrac{\partial \mathbf{D}}{\partial \mathbf{a}} &= \mathbf{0} \\ \tfrac{\partial \mathbf{D}}{\partial \mathbf{w}} &= \mathbf{0}. \end{aligned} \tag{5.8}$$

To simplify the analysis, we define vector $\mathbf{b}$ as the concatenation of $\mathbf{a}$ and $\mathbf{w}$:

$$\mathbf{b} = [a_1, a_2, ..., a_c, w_1, w_2, ..., w_n],$$

and symmetric matrix $\mathbf{B_i}$:

$$\mathbf{B_i} = \begin{pmatrix} 0 & \mathbf{A_i}/2 \\ \mathbf{A_i}^T/2 & 0 \end{pmatrix}.$$

We have

$$\mathbf{a}^T\mathbf{A_i}\mathbf{w} = \mathbf{b}^T\mathbf{B_i}\mathbf{b},$$

then Eq. 5.8 becomes:

$$\begin{aligned} \frac{\partial \mathbf{D}}{\partial \mathbf{b}} &= \sum_{i=1}^{N} \frac{\partial (y_i - \mathbf{b}^T\mathbf{B_i}\mathbf{b})^2}{\partial \mathbf{b}} \\ &= \sum_{i=1}^{N} 2(y_i - \mathbf{b}^T\mathbf{B_i}\mathbf{b}) \cdot 2\mathbf{b}^T\mathbf{B_i} \\ &= 4\sum_{i=1}^{n} (y_i - \mathbf{b}^T\mathbf{B_i}\mathbf{b})\mathbf{b}^T\mathbf{B_i} \\ &= 4(\sum_{i=1}^{n} y_i\mathbf{b}^T\mathbf{B_i} - \sum_{i=1}^{n} \mathbf{b}^T\mathbf{B_i}\mathbf{b}\mathbf{b}^T{\mathbf{B_i}}^T) \\ &= 0 \end{aligned}$$

This is a third order equation array, for which the analytical solution is unknown. We have not found any existing publication related to this particular problem. However, we can certainly apply general nonlinear numerical optimization methods (such as quasi newton methods) to solve this problem. We present here a solution with very simple form.

Note that in Eq. 5.7, if we fix $\mathbf{a}$, then optimization $\mathbf{w}$ is a linear regression problem. Similarly, we can find optimal estimate of $\mathbf{a}$ with linear regression, by fixing $\mathbf{w}$. So if we fix $\mathbf{a}$ and $\mathbf{w}$ alternately and calculate the other using linear regression, the process will converge to at least a local minimum, as we will show a little later. The details are shown in algorithm 1.

5.2.2 Convergence

First of all, the above algorithm is guaranteed to converge to a local minimum, if we assume $(V^TV)^{-1}$ and $(U^TU)^{-1}$ always exist. Denote the values of $\mathbf{a}, \mathbf{w}$ in the ith iteration with $\mathbf{a^i}, \mathbf{w^i}$. Note that linear regressions find the solution with the

Algorithm 1 BILINEARREGRESSION(S,Y,MAP) Iteratively find HRF and weights of regressors using Criss-Cross regression

$\mathbf{a} \leftarrow \mathbf{0}$; $\mathbf{w} \leftarrow \mathbf{1}$
$iterations \leftarrow 0$
Build SS from S
while $\|\mathbf{a} - \mathbf{a_{old}}\|_2 >$ NormThres and $iterations <$ IterThres **do**
  /* Estimate a using w*/
  $U \leftarrow (\mathbf{S_1 w}, \mathbf{S_2 w}, \ldots, \mathbf{S_N w})^T$
  $\mathbf{a_{old}} \leftarrow \mathbf{a}$
  $\mathbf{a} \leftarrow (U^T U)^{-1} U^T Y$
  /* Estimate w using a*/
  $V \leftarrow (\mathbf{S_1}^T \mathbf{a}, \mathbf{S_2}^T \mathbf{a}, \ldots, \mathbf{S_N}^T \mathbf{a})^T$
  if MAP **then**
    $\mathbf{w} \leftarrow (V^T V + var\Sigma^{-1})^{-1} V^T Y$
  else
    $\mathbf{w} \leftarrow (V^T V)^{-1} V^T Y$
  end if
  $\mathbf{w} \leftarrow \mathbf{w} / |\mathbf{w}|$
  $iterations \leftarrow iterations + 1$
end while
return $\mathbf{w}$

least square error. So we have

$$\mathbf{D}(\mathbf{a_1}, \mathbf{w_1}) \geq \mathbf{D}(\mathbf{a_2}, \mathbf{w_1}) \geq ... \geq \mathbf{D}(\mathbf{a_i}, \mathbf{w_i}) \geq \mathbf{D}(\mathbf{a_{i+1}}, \mathbf{w_i}) \geq \mathbf{D}(\mathbf{a_{i+1}}, \mathbf{w_{i+1}}) \geq ...$$

In other words, the objective function will monotonically decrease. Due to the fact that $\mathbf{D}(\mathbf{a}, \mathbf{w})$ has a lower bound of 0, this algorithm can not decrease forever, thus it has to converge to some local minimum, on which Eq. 5.8 holds.

Note that there are multiple local minima. A trivial case is that: for an optimal value set of $(\mathbf{a}, \mathbf{w})$, $(-\mathbf{a}, -\mathbf{w})$ is also a local minimum. We present an empirical study to examine the convergence in 11.2.

5.2.3 Notes

Variants of the "alternating regression" method presented in this section were applied in matrix factorization problems ([Gabriel and Zamir, 1979, Gabriel, 1998, Kim and Park, 2007]). Studies about the convergence rate were not included in

those papers.

Chapter 6
A Nonlinear Hemodynamic Response Model

The models we described so far, are all Linear Time Invariant (LTI) models. i.e., a response is represented as the convolution of a HRF with a stimulus time series. These models have proven to be a useful approximation of real hemodynamics, and have many important applications in neuroscience. In addition, several non-linear models have been developed. The most important ones among them are the balloon model [Buxton et al., 1998] and Volterra kernels[Friston et al., 2000]. The balloon model has several parts, which are intended to represent aspects of physiology. The Volterra model starts from a very general formulation of a history-dependent response. They provide insights into the interactions among different factors, and explain the observed hemodynamic behavior more precisely. However, these models are complicated have too many parameters to adjust. This is inevitable in attempting to achieve full understanding the phenomenon, but it is too expensive for indexing and retrieval for massive data collections.

We examine [1] a simple non-linear model. This model shares a philosophy with balloon model, but it is simple and cheap enough for whole brain processing. Note that the term "nonlinear" only describes the non-linearity when generating the hypothesized response. This response is still used in GLM framework to do statistical inference. We thus still include this section in this chapter.

[1] This model is proposed by Paul Kantor.

6.1 Lagged, Limited First Order Model (LLFOM)

Suppose we are studying a time series of a voxel $y_t, t \in [1, N]$, which contains the hemodynamic response to the stimulus $s_t, t \in [1, N]$. We assume that $y_t = \hat{y}_t + \Delta + \epsilon$, where $\hat{y}_t$ is the modeled response to the stimulus s_t, Δ is the constant term. The lagged, limited first order model (LLFOM) is given by Eq. 6.1,

$$\frac{d\hat{y}}{dt} = a \cdot s(t-\tau)(y_{max} - \hat{y}_t) - b \cdot \hat{y}_t, \tag{6.1}$$

where $\tau \geq 0, y_{max} \geq 0, b \geq 0$. The rate of change of $\hat{y}$ consists of two parts. The first term is the "positive" response, which is proportional to stimulus at some earlier moment $t - \tau$, with a strong factor a. This term is limited by the factor $y_{max} - \hat{y}_t$. That is, the closer $\hat{y}_t$ is to its upper limit of its ability, the harder it is for the response to increase. The second term is an exponential decay. The decrease is proportional to current response. Clearly, this is a linear ordinary differential equation, which means the response approaches the stable state (0 in this formulation) with an exponential rate. The four parameters a, τ, y_{max}, b are all non-negative.

For analyses, we regroup the terms and have:

$$\frac{d\hat{y}}{dt} = As(t-\tau) - Bs(t-\tau)\hat{y}_t - C\hat{y}_t, \tag{6.2}$$

where $A = a \cdot y_{max}$, $B = a$, $C = b$.

6.2 Model fitting

Model fitting seeks optimal values of A, B, C, τ and a constant Δ, so that

$$D = \sum_{i=1}^{N} (y_i - \hat{y}_i - \Delta)^2 \tag{6.3}$$

is minimized. This is a nonlinear optimization problem.

Non-linear programming functions can be found in general purpose software such as Microsoft Excel or Matlab optimization toolbox, or in separate packages

such as OPT++ [Meza, 1994] or L-bfgs-B [Zhu et al., 1997]. While the former have very convenient user interfaces, they are not suitable for the massive computations we need for fMRI image processing. After comparing several available packages we have available, we chose L-bfgs-B. L-bfgs-B is a FORTRAN77 software package that implements a quasi-newton algorithm BFGS [Nocedal and Wright, 1999] with bounds constraints. Although the I/O operations is relatively harder to handle, it gives more stable results with a faster speed. The details of BFGS algorithm with bound restraints [Nocedal and Wright, 1999], will not be covered here. Like many other nonlinear optimization methods, BFGS requires the user to provide both the *function value* and the *gradient* at every iteration, in order to define the searching direction.

The objective function value D can be calculated explicitly. Given current estimates of A, B, C, τ and the constant Δ, we can approximate $\hat{y}_t$ by the first order Taylor expansion:

$$\frac{d\hat{y}_{t-1}}{dt} \approx \hat{y}_t - \hat{y}_{t-1}, \hat{y}_0 = 0.$$

Eq. 6.3 is then applied to calculate D.

There are two options for partial derivatives. First, we can apply methods that are purely numerical. For example, we can compute $\frac{\partial D(A,B,C,\tau,\Delta)}{\partial A}$ as follows:

$$\frac{\partial D(A, B, C, \tau, \Delta)}{\partial A} = \frac{D(A + \epsilon, B, C, \tau, \Delta) - D(A, B, C, \tau, \Delta)}{\epsilon}.$$

In this approach, ϵ needs to be tuned carefully. Large ϵ increase the error in the estimate, while too small ϵ may cause loss of numerical precision.

The partial derivatives with respect to other parameters can be computed in the same way, except for τ, which is an integer. Since we only consider τ in a limited range (30 seconds), we can conduct a grid search on τ.

We can also use "semi-analytical" gradient, whose calculation does not depend on the choice of ϵ. Let $D = \sum_{t=1}^{N} \delta(t)$, where $\delta(t) = (y_t - \hat{y}_t - \Delta)^2$. For a parameter

$p \in \{A, B, C, \Delta\}$,

$$\frac{\partial D}{\partial p} = \sum_{t=1}^{N} \frac{\partial \delta(t)}{\partial p}, \tag{6.4}$$

where

$$\frac{\partial \delta(t)}{\partial p} = \begin{cases} \sum_{t=1}^{N} \frac{\partial \delta(t-1)}{\partial p} + \frac{\partial (\frac{d\hat{y}_t}{dt})}{\partial p} & t \geq 2 \\ 0 & t = 1 \end{cases} \tag{6.5}$$

$\frac{\partial (\frac{d\hat{y}_t}{dt})}{\partial p}$ for $p = A, B, C, \Delta$ are listed below:

$$\begin{array}{rcl} \frac{\partial (\frac{d\hat{y}_t}{dt})}{\partial A} & = & s(t-\tau) - (Bs(t-\tau) + C)\frac{\partial (\frac{d\hat{y}_{t-1}}{dt})}{\partial A} \\ \frac{\partial (\frac{d\hat{y}_t}{dt})}{\partial B} & = & -s(t-\tau)\hat{y}_{t-1} - (Bs(t-\tau) + C)\frac{\partial (\frac{d\hat{y}_{t-1}}{dt})}{\partial B} \\ \frac{\partial (\frac{d\hat{y}_t}{dt})}{\partial C} & = & -\hat{y}_{t-1} - C\frac{\partial (\frac{d\hat{y}_{t-1}}{dt})}{\partial C} \\ \frac{\partial (\frac{d\hat{y}_t}{dt})}{\partial \Delta} & = & 0 \end{array} \tag{6.6}$$

The details of this implementation are in Algorithm 2.

6.3 Notes

This chapter includes parts of [Bai and Kantor, 2007, Bai et al., 2007b], which are co-authored by Bing Bai, Paul Kantor, Ali Shokoufandeh, Nicu Cornea, and Deborah Silver.

Algorithm 2 NONLINEARREGRESSION(S, Y, A_0, B_0, C_0, T) Estimate parameters in the nonlinear model, assume τ is fixed

/* A_0, B_0, C_0 are initial values of A, B, C */

$A \leftarrow A_0, B \leftarrow B_0, C \leftarrow C_0, \Delta \leftarrow 0$
$A_{old} \leftarrow \infty, B_{old} \leftarrow \infty, C_{old} \leftarrow \infty, \Delta_{old} \leftarrow \infty$
while $(A - A_{old})^2 + (B - B_{old})^2 + (C - C_{old})^2 + (\Delta - \Delta_{old})^2 > T$ **do**
  $A_{old} \leftarrow A, B_{old} \leftarrow B, C_{old} \leftarrow C, \Delta_{old} \leftarrow \Delta$
  $D_A \leftarrow 0, D_B \leftarrow 0, D_C \leftarrow 0, D_\delta \leftarrow 0$ // components of gradients: D_A means $\frac{\partial D}{\partial A}$, similar for D_B, D_C, D_Δ.
  $\delta_{t,A} \leftarrow 0, \delta_{t,B} \leftarrow 0, \delta_{t,C} \leftarrow 0$ // $\delta_{t,A}$ means $\frac{\partial \frac{d\hat{y}_t}{dt}}{\partial A}$, similar for $\delta_{t,B}, \delta_{t,C}, \delta_{t,\Delta}$.
  $obj \leftarrow 0$
  $y_c \leftarrow 0$ // current value of y
  for $i = 1$ to N **do**
    $idx_s \leftarrow i - t$ // the index for stimulus
    if $idx_s < 0$ **then**
      $s_{t-\tau} \leftarrow 0$
    else
      $s_{t-\tau} \leftarrow S(idx_s)$
    end if
    $y_{old} \leftarrow y_c$
    $dy \leftarrow As_{t-\tau} - Bs_{t-\tau}y_c - Cy_c$
    $y_c \leftarrow y_c + dy$
    if $y_c < 0$ **then**
      $y_c \leftarrow 0$ // maintain non-negativity of estimate
    end if
    $ct \leftarrow y_c - \Delta - \mathrm{Y(t)}$ // common term
    $obj \leftarrow obj + ct^2$ // update objective value
    /* now calculate gradient */
    $\delta_{t,A} \leftarrow \delta_{t,A} + s_{t-\tau} - (B \cdot s_{t-\tau} + C)\delta_{t,A}$
    $\delta_{t,B} \leftarrow \delta_{t,B} - s_{t-\tau} \cdot y_{old} - (B \cdot s_{t-\tau} + C)\delta_{t,B}$
    $\delta_{t,C} \leftarrow \delta_{t,C} - y_{old} - C \cdot \delta_{t,C}$
    $D_A \leftarrow D_A + 2ct \cdot \delta_{t,A}$
    $D_B \leftarrow D_B + 2ct \cdot \delta_{t,B}$
    $D_C \leftarrow D_C + 2ct \cdot \delta_{t,C}$
    $D_\Delta \leftarrow D_\Delta - 2ct$
  end for/* Estimate (A, B, C, Δ) with BFGS*/
  $(A, B, C, \Delta) \leftarrow BFGS(obj, D_A, D_B, D_C, D_\Delta)$

end while
return **w**

Chapter 7

Features based on Independent Component Analysis

We adopt here the Probabilistic Independent Component Analysis (PICA) model implemented in FSL, to generate independent components. The dimension of PICA result (the number of components) depends on experimental conditions and the length of the time series. In our studies, this number varies from less than 20 to more than 100. Many of these components are not task-related. Our goal is to identify task-related components, or at least reduce the number of potentially task-related components, without using the stimulus time series. Can we distinguish the task-related components? In [Hu et al., 2005], the components are considered more task related if their time courses have stronger correlation to known stimulus time series. However, for datasets with more than one cognitive task, it may be difficult to separately tag the independent components. See 3.3.2 for details of ICA.

Here we propose a heuristics to select task related components, based on the frequency spectra. We will shown later that, with this method we achieve higher and more stable retrieval performance than alternative approaches.

7.1 New component selection based on low mean frequency

We know that PICA decomposes a temporal-spatial matrix into "independent components" (independent spatial maps and their corresponding time courses). Here we propose a simple heuristic based on the spectrum of the time course. In

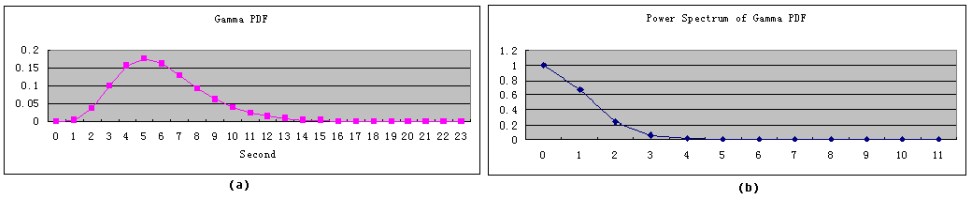


Figure 7.1: (a) gamma function. (b) The power spectrum of the gamma function

the GLM, a task-related time course is modeled with the convolution of HRF and the stimulus time series. As stated in earlier sections , it is assumed to be similar to a gamma PDF (Figure 7.1). In Figure 7.1 we see that the mass of energy for a gamma PDF is on the low end of the spectrum, implying that the gamma PDF is a low-pass filter. By the convolution theorem:

$$F(f * g) = F(f) \cdot F(g)$$

,we know that the energy of the explanatory variable must also tend to stay in the relatively low end of the spectrum. We use the following metric, which we call *expected frequency* to indicate the energy tendency:

$$E(f) = \frac{\sum_{i=1}^{N} i \cdot A_i}{\sum_{i=1}^{N} A_i}$$

$A_i = |A(w)|$ is the amplitude of ith element in the discrete power spectrum.

Another reason to select low frequency components is that, most fMRI experiment use a low frequency design for stimuli.

One example of task related component is shown in Figure 7.2. On the top is the brain activation map, on the bottom is its corresponding time course and stimulus.

7.2 Notes

This chapter includes parts of [Bai et al., 2007c], which are co-authored by Bing Bai, Paul Kantor, Ali Shokoufandeh and Deborah Silver.

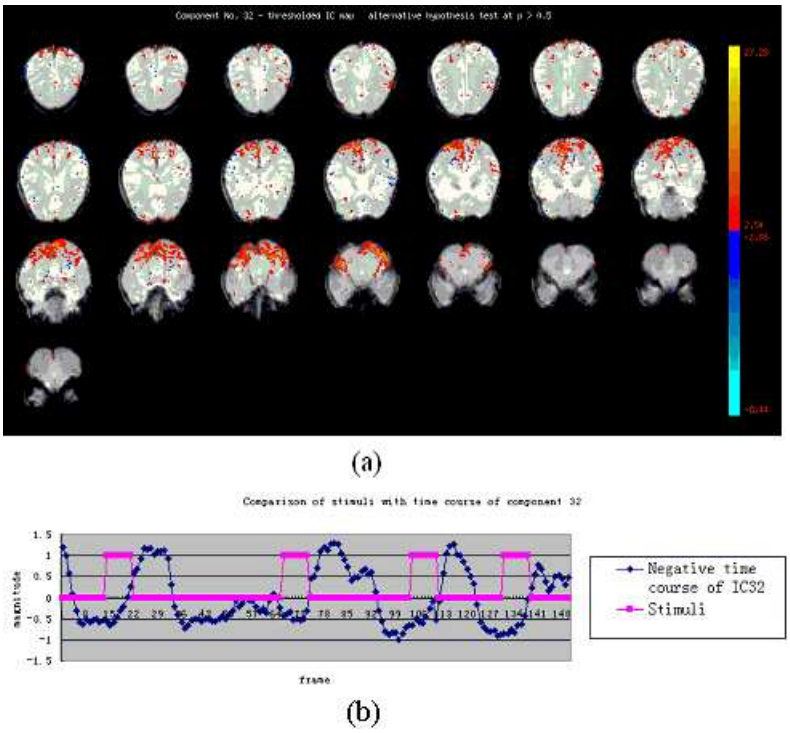


Figure 7.2: An example component of ICA. (a) is the spatial map for this component. (b) is the time course corresponding to this component.

Chapter 8
Information Retrieval Measures

To accomplish object recognition, features must be compared to a model or to each other after extraction. The measure of similarity (or, on the contrary, distance) is generated for all pairs of objects. We refer to this process as "matching" in this context. That is, because we expect correspondences between the feature sets from two related objects, we try to match features in one set to those in another set. At the same time, we may infer a distance between them.

8.1 Cosine and Euclidean-like distance

In this thesis, we define "similarity" as any measure inverse tothe "distance". We will simply use the term "X similarity" as the negative of the "X distance", in regarding to the similarity/distance type X.

Given a positive integer p, p-norm distance Euclidean distance between two vector $\mathbf{a}$ and $\mathbf{b}$ is defined as:

$$d_p(\mathbf{a}, \mathbf{b}) = (\sum_{i=1}^{n} |a_i - b_i|^p)^{\frac{1}{p}}. \tag{8.1}$$

2-norm distance is also called *Euclidean distance*, while 1-norm distance is called Manhattan distance.

Mahalanobis distance is an extension on the Euclidean distance. Given a set of n samples $\mathbf{x}^i = (x_1^i, x_2^i, ..., x_m^i), i \in [1, n]$ in m-dimension space, the mean value of each coordinate $\mu = (\mu_1, \mu_2, ..., \mu_m)$ can be calculated as: $\mu_p = \frac{1}{n} \sum_{i=1}^{n} x_p^i$, and

the covariance matrix for the m coordinates is:

$$\mathbf{S} = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2m} \\ \dots & \dots & \dots & \dots \\ s_{m1} & s_{m2} & \dots & s_{mm} \end{pmatrix},$$

where $s_{pq} = \frac{1}{n}\sum_{i=1}^{n}(x_p^i - \mu_p)(x_q^i - \mu_q)$. Then the Mahalanobis distance between two points $\mathbf{u}$ and $\mathbf{v}$ is:

$$d_{ma}(\mathbf{u}, \mathbf{v}) = \sqrt{(\mathbf{u} - \mathbf{v})^T \mathbf{S}^{-1} (\mathbf{u} - \mathbf{v})}. \tag{8.2}$$

When $\mathbf{S}$ is a diagonal matrix (i.e., all the coordinates are not correlated), the Mahalnobis distance is simplified to:

$$d_{ma} = \sqrt{\sum_{i=1} m \frac{u_i - v_i}{s_{ii}}}. \tag{8.3}$$

Intuitively, Mahalanobis distance normalizes each coordinate to prevent the difference between the coordinates with small variance being overwhelmed by the coordinates with large variance.

Another distance measure for two vectors is "Cosine". It is defined as the negative of the inner product of the two normalized vectors:

$$d_{cosine}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^T \mathbf{v}}{\sqrt{\mathbf{u}^T \mathbf{u}} \sqrt{\mathbf{v}^T \mathbf{v}}} \tag{8.4}$$

8.2 Simple Overlap

The method we call "overlap" is also called "Jaccard distance" in other literatures. Specifically, the distance between two set of selected voxels is simply the size of their overlap divided by the size of their union: $similarity(A, B) = \|A \cap B\| / \|A \cup B\|$. The reason we are using this new term is that we like to use "overlap" to describe several methods sharing the same fundamental idea. This basic method

is "simple overlap" or "overlap", while the slightly more sophisticated variant is called "fuzzy overlap" in next section.

In this method, the retrieved datasets are ranked according to the size of their overlap with the activated voxels in the query image. To maintain the analogy to "terms" in text retrieval, we refer to these as the "query voxels". For instance, a candidate dataset whose representation contains 50 of the query voxels will outrank a dataset that contains only 49 of query voxels.

Since activation is here binarized, the overlap method is also a variation of the cosine measure for vector similarities. The thresholded t-maps can be considered as sparse vectors, with value 1 for activated voxels, and 0 for the rest. It can be shown that if the number of "1"s (that is, the number of selected voxels) is the same for vectors A, B, C and D, then

$$cos(A, B) > cos(C, D) \Leftrightarrow overlap(A, B) > overlap(C, D)$$

Thus, for the binarized representation with exactly the same number of active voxels, overlap and the cosine measure are equivalent in the sense of retrieved ranked lists.

8.3 Fuzzy Overlap

Now consider the following case. A specific voxel r is active in two different brain images A and B. However, the normalized coordinates of voxel r_a and voxel r_b do not overlap in standard space, due to differences in the shape of the brains, although they might be close. If we only index the datasets with the exact activated voxel (in this case, A is in r_a's entry, B is in r_b's entry), then the retrieval system will miss B if the query is A. To address this problem, we index a dataset A not only with its activated voxel r_a, but also with the *neighbors* of r_a. Specifically, we say that a voxel $u = (x_1, y_1, z_1)$ is a neighbor of voxel $v = (x_2, y_2, z_2)$ if the L-infinity-norm (L_∞) $|u - v|_\infty < R_f$. R_f is called

the *fuzziness radius*. If R_f is set too small, some related datasets may be missed. If R_f is set too large, false alarms increase, and precision will drop. Figure 8.1 shows a 2D case with fuzziness radius 1. The voxel marked 8 is active in image A, so A is put in the index entries of voxel 8 and of its neighbors.

This method, to which we refer as *fuzzy overlap*, is similar to the standard morphology-based operators used in image processing [Heijmans, 1994].

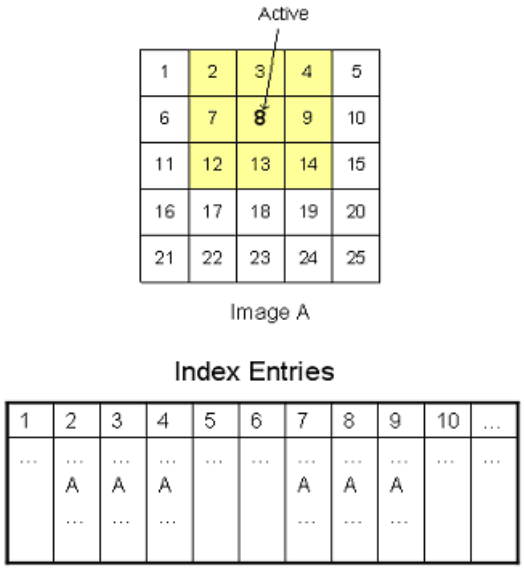


Figure 8.1: One example of fuzziness radius 1. Voxel 8 in Image A is active, A is indexed with voxel 8 and its neighbors

8.4 Latent Semantic Indexing

Latent Semantic Indexing (LSI) [Deerwester et al., 1990] is a technique to automatically discover similarities between terms and documents, and to apply this information to information retrieval. Suppose for n documents, the total number of terms is $m \geq n$, we first build an $m \times n$ term-document matrix M. In this matrix, the element at location (i, j) is the number of occurrences of term i in document j. Singular value decomposition (SVD) [Nash, 1990] can decompose

this matrix into the form:

$$M = U \cdot S \cdot V^T, \tag{8.5}$$

where U (a m by r matrix) contain the orthonormal eigenvectors of MM^T: $\mathbf{u}_1\mathbf{u}_2...\mathbf{u}_r]$, V (a n by r matrix) contains the orthonormal eigenvectors of M^TM: $\mathbf{v}_1\mathbf{v}_2...\mathbf{v}_r$. Both MM^T and M^TM share the same set of eigenvalues, which are the diagonal elements in diagonal matrix S. SVD can also be written in the form of:

$$M = \sum_{i=1}^{r} s_i \mathbf{u}_i \mathbf{v}_i^T,$$

where r is the rank of M, and each set of $s_i, \mathbf{u}_i\mathbf{v}_i, i \in [1, r]$ is called a "triplet". Usually the eigenvalues are arranged in decreasing order, i.e. $s_1 \geq s_2 \geq ... \geq s_r$. These eigenvalues are called "singular values".

In LSI, U, is called the *term matrix*. in which each column is interpreted as a "latent semantic component"; each element in the column is the weight of the corresponding term in this component. V is called the document matrix. The ith column of V contains the weights of the "latent semantic components" in document i.

In the case of text, when a query is received, a term vector q for the query is built first; each element in this vector is the number of occurrence of the corresponding term. Then this term vector is transformed into semantic space by projection onto the semantic components:

$$\hat{q} = q^T U S^{-1}. \tag{8.6}$$

The similarity between the query and the ith document is defined as the cosine of $\hat{q}$ and V_i.

$$\cos(\hat{q}, V_i) = \frac{\sum_j (\hat{q}_j \cdot V_{ij})}{\sqrt{\sum_j \hat{q}_j^2 \sum_j V_{ij}^2}} \tag{8.7}$$

As noted, 1 percent of the voxels, those with largest t-values are selected to be "activated". If a voxel is activated, the corresponding value in the term vector

is set to 1, otherwise it is set to 0. The standard routine of LSI (equation 8.5, 8.6, 8.7) is applied after that.

8.5 Principal Component Analysis

Principal Component Analysis (PCA) is not a typical technique used in text retrieval, but is a popular method used in image data reduction. Mathematically, PCA is very close to LSI in its mathematical nature. They both rely on SVD, and have very similar formulation. The only difference is that PCA works on a matrix in which the mean value is removed from every dimension, while LSI works on the raw matrix. For example, if we want to apply PCA to a term-document matrix, we need to demean the term frequency for each term first. We demonstrate the difference with Figure 8.2.

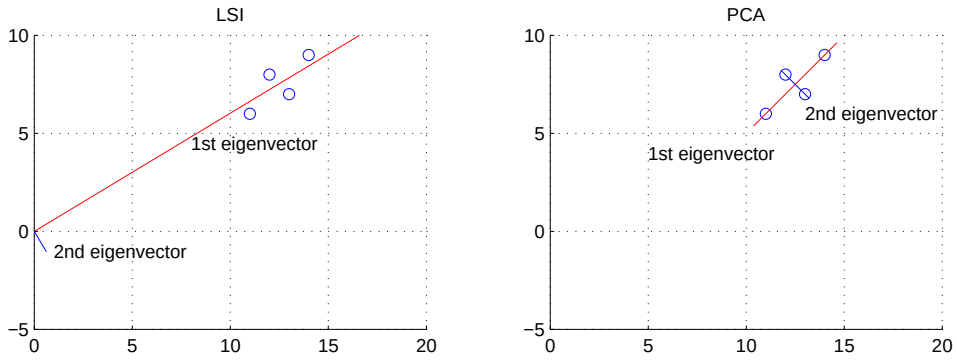


Figure 8.2: The difference between LSI and PCA. The circles are samples. The two lines represent two eigenvectors. The ratio between the lengths of two lines is equivalent to the ratio between the first eigenvalue and the second eigenvalue.

Figure 8.2 shows a case with 4 samples, (11, 6), (14,9), (13,7), and (12,8). When looking for the first eigenvector in LSI, we apply SVD to the raw sample-coordinate matrix:

$$\begin{pmatrix} 11 & 14 & 12 & 13 \\ 6 & 9 & 8 & 7 \end{pmatrix}.$$

If we show the first eigenvector (the one with the largest eigenvalue) with the start end at the origin, it will roughly pass the "center of the mass" of all samples. In case of PCA, we apply SVD to the matrix with the mean value of x and y coordinates removed:

$$\begin{pmatrix} -1.5 & 1.5 & 0.5 & -0.5 \\ -1.5 & 1.5 & -0.5 & 0.5 \end{pmatrix}.$$

The first eigenvector is along the direction where the projections of the samples yield largest variance. We see that LSI and PCA can generate very different results when mean values are large.

8.6 TFIDF

Under TFIDF [Salton and Buckley, 1988] the weight of a term is the product of two parts: TF(term frequency) and IDF (inverse document frequency). For each term t, TF indicates the importance of t in a given document d, while IDF indicates the general importance of t. TFIDF has many variations, and a popular form is the following:

$$tf(t,d) = \frac{n(t,d)}{\sum_k n(k,d)} \tag{8.8}$$

$$idf(t) = \log(\frac{D}{T}) \tag{8.9}$$

$$tfidf(t,d) = tf(t,d)idf(t) \tag{8.10}$$

$n(t,d)$ is the number of times term t occurs in document d. $tf(t,d)$ is $n(t,d)$ divided by the document length. D is the total number of documents in the collection. T is the number of documents containing the term t.

We have to translate this to brain images. Unlike a term in a document, an activated voxel appears only once in any brain image. So if we mimic the behavior of term frequency, the voxel/term frequency is always 1. To give a finer

representation, we can replace the term frequency with the t-value to indicate the local importance of the term.

IDF translates naturally to fMRI images. If a voxel is activated in too many images, it is obvious that this voxel is not a discriminant feature. One source of such voxels are vessels, especially arteries. Oxygenated blood flow has to pass through these vessels to reach the actual activated regions.

Once the weights for all the voxels are known, a sparse vector is built for every thresholded t-map; the t-value for each voxel is its weight. Again the similarity measure between two thresholded t-maps is defined as the cosine of the vectors.

8.7 Notes

This chapter includes parts of [Bai et al., 2006, 2007a], which are co-authored by Bing Bai, Paul Kantor, Nicu Cornea, and Deborah Silver.

Chapter 9
Bipartite Graph Matching

9.1 Unweighted Bipartite Graphs

Before we move on, we need to define some important terms. If a matching in a graph can not be enlarged by adding an edge, it is called *maximal matching*. The largest possible matching is called a *maximum matching*. If every vertex of this graph is in a matching, that match is called a *perfect matching*. If a vertex is incident to the edges of a matching M, it's called *M-saturated*, otherwise it's *M-unsaturated*. Two key concepts need to be defined here:

Definition 1 (M-alternating path) *Given a matching M, and an M-alternating path is a path that alternates between edges in M and edges not in M.*

Definition 2 (M-augmenting path) *If the end points of M-alternating path are unsaturated by M, the path is called a M-augmenting path.*

Whether a matching is an maximum matching can be decided with the theorem below:

Theorem 1 (Berge's theorem) *A matching M is maximum matching in graph G iff there is no M-augmenting path in G.*

Not every graph can have a perfect matching. Whether a graph has a perfect matching can be decided with theorem 2 [Bondy and Murty, 1976].

Definition 3 (Neighbor set) *For any set S of vertices in G, the "neighbor set" of S in G is defined as the set of all vertices adjacent to vertices in S. This set is denoted by $N_G(S)$.*

Theorem 2 (Hall's theorem) *Let G be a bipartite graph with bipartition (X, Y). Then G contains a perfect matching iff $|N_G(S)| \geq |S|$ for all $S \in X$.*

Definition 4 (Symmetric difference) *If M and M are matchings, then $M \Delta M = (M - M) \cup (M - M)$ is called symmetric difference between M and M.*

9.1.1 Finding perfect matching in unweighted bipartite graphs

Given a bipartite graph, we can use the so-called Hungarian method [Edmonds, Bondy and Murty, 1976] to find a perfect matching. Figure 9.1 shows the flow chart of this method.

$N(S)$ is the neighbor set of set S. $M \Delta E(P)$ means "switch the matching edges and unmatching edges in M-augmenting path P to make a bigger matching". The idea of Hungarian method is this: starting from any matching, we try to grow a "M-alternating tree" from every unmatched vertex u. A "M-alternating tree" rooted at u is a subgraph of G, with two bipartite set S and T, such that for any other vertex v in this subgraph, the unique path between u and v is an M-alternating path, see Figure 9.2. Originally, S only contains an unmatched vertex u, and T is empty. We always check the neighbors of S which are not in T. If such a neighbor y is already matched to a vertex z, we add y to T, and z to S, so T always contains the "matched vertices" of S. If y is not matched, that means there is a M-augmenting path from u to y, and we switch the "matching flags" of edges in this path to make M a bigger matching, then we pick another unmatched vertex from G, and repeat this process.

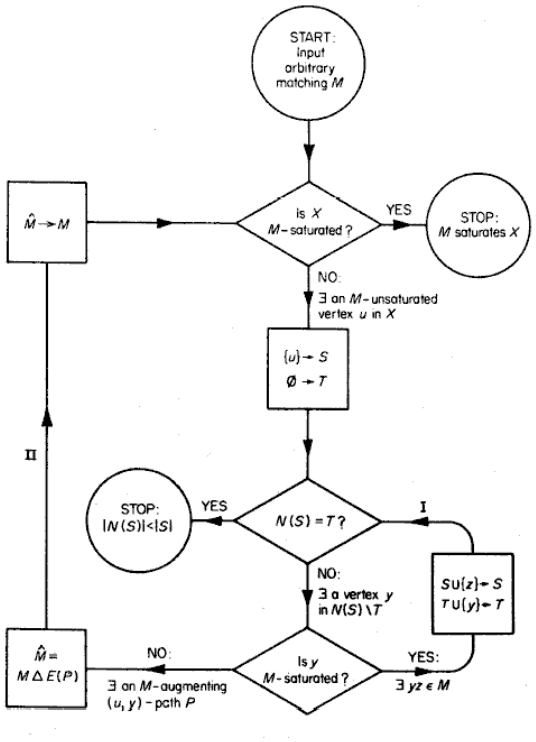


Figure 9.1: From [Bondy and Murty, 1976]
. A flow chart of Hungarian method to find perfect matching.

There can be two results of this algorithm, either a perfect matching is found, or an M-augmenting path is not found for an unmatched u, on which the algorithm reports "no perfect matching".

9.2 Weighted bipartite graphs and the Optimal assignment problem

Suppose there are n workers and n jobs, different workers can earn, for us, different profit doing different jobs, and each worker can only do one job. How can we get the maximum profit by assigning jobs to workers?

The above problem is called optimal assignment problem. This problem can be conveniently expressed with a weighted bipartite graph: the workers are in

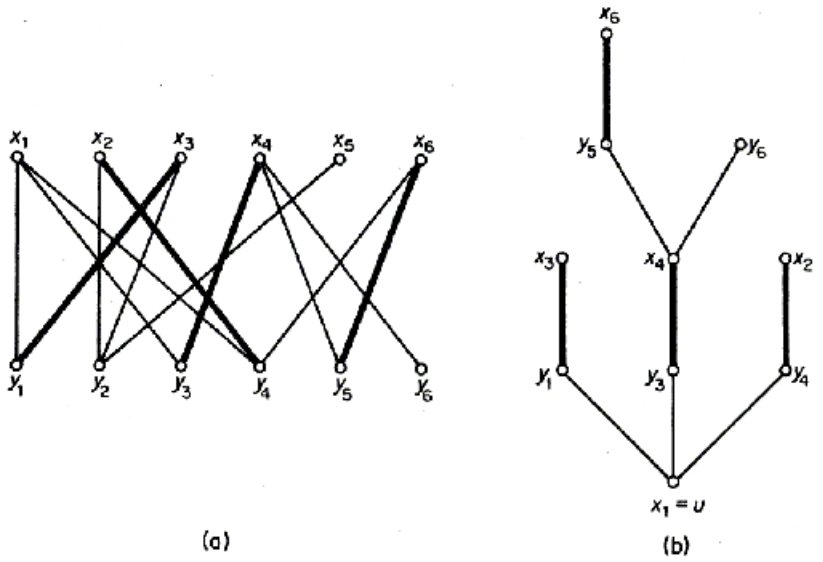


Figure 9.2: From [Bondy and Murty, 1976]
. (a) A matching M in G. (b) an M alternating tree in G.

set S, the jobs are in set T, and the profits of the worker-job pair are the weight of the edge connecting the vertices representing the worker and the job. Thus, this optimal assignment problem can be restated as optimal matching problem in language of graph theory as following:

Definition 5 (Optimal matching) *For a matching M in weighted bipartite graph, if there is not a matching M' such that the sum of the weights in M' is greater than the sum of weights in M, then M is called a optimal matching.*

This problem is addressed with the Kuhn-Munkres algorithm Kuhn [1955], Munkres [1957] which is based on a labeling system and Hungarian method we introduced before.

Definition 6 *A feasible vertex labeling is a real valued function l on vertices in G such that, for each bipartite pair $x \in X$ and $y \in Y$,*

$$l(x) + l(y) \geq w(xy)$$

A feasible label can be easily found with the following scheme:

$$\left.\begin{array}{rl} l(x) = \max_{y \in Y} w(xy) & if \quad x \in X \\ l(y) = 0 & if \quad y \in Y \end{array}\right\}. \tag{9.1}$$

If l is a feasible vertex labeling, we denote by E_l the set of edges where the sum of labels of both ends equals the weight.

$$E_l = \{xy \in E | l(x) + l(y) = w(xy)\} \tag{9.2}$$

The graph G_l with edge set E_l and its vertices is called an equality subgraph.

Theorem 3 *If an equality subgraph G_l contains a perfect matching M, M is an optimal matching of G.*

The Kuhn-Munkres algorithm can be stated as following:

1. Find any feasible vertex labeling l, find G_l, and choose an arbitrary matching M in G_l.

2. If M is a perfect matching, then M is an optimal matching, stop. Otherwise, let u be an unmatched vertex, set $S = \{u\}$ and $T = \emptyset$.

3. Let $N_{G_l}(S)$ be the neighbors of S in equality subgraph G_l. if $T \subset N_{G_l}(S)$, go to step 4. Otherwise, let $\alpha_l = \min_{x \in S, y \notin T}\{l(x) + l(y) - w(xy)\}$, and we recalculate a feasible vertex labeling $\widehat{l}$ as follows:

$$\hat{l} = \begin{cases} l(v) - \alpha_l & : \quad v \in S \\ l(v) + \alpha_l & : \quad v \in T \\ l(v) & : \quad otherwise \end{cases} \tag{9.3}$$

   , and recalculate equality subgraph G_l.

4. This step is similar to the Hungarian method. Choose a vertex y in N_{G_l} T. If y is already matched with z, $S \leftarrow S \cup \{z\}$, $T \leftarrow S \cup \{y\}$ and go to step 3. if y is not matched, then there is an M-augmenting path in G_l, switch the matching flag along the path, and go to step 2.

The major difference from the Hungarian method is in step 3. Once a M-alternating tree can't grow in current equality subgraph, we introduce a new edge into equality subgraph, and redo the procedure. Every edge will, sooner or later, be added into the equality subgraph if necessary, and a solution is guaranteed.

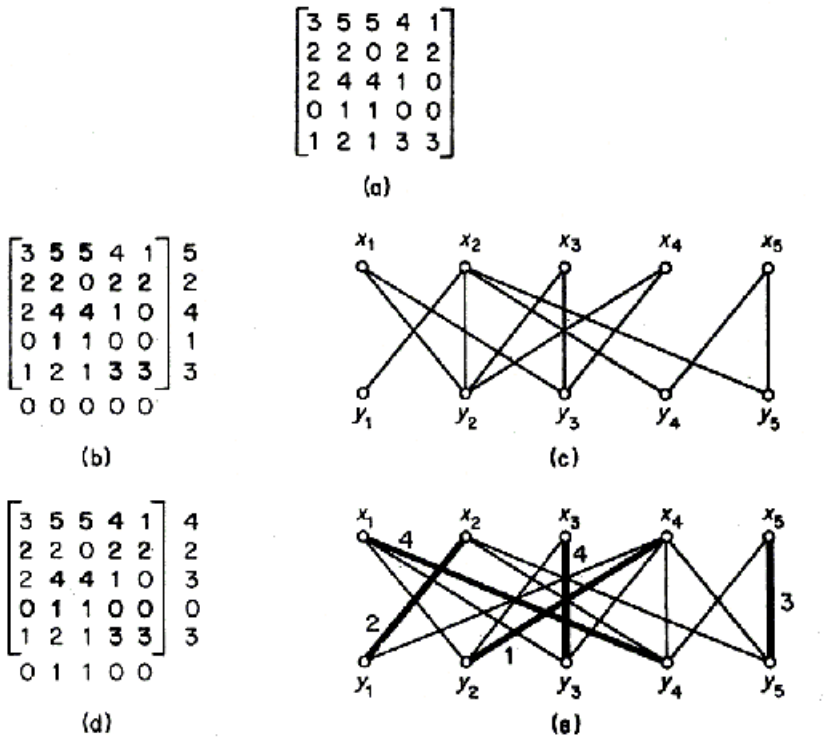


Figure 9.3: From [Bondy and Murty, 1976]. Illustration of the Kuhn-Munkres algorithm

To illustrate the Kuhn-Munkres algorithm, it's convenient to represent the weights of edges by a matrix $W = [w_{ij}]$, where w_{ij} is the weight of $x_i y_j$ in G. See Figure 9.3a. We give an initial labeling according to equation 9.1, which is shown in Figure 9.3b. The label is shown on the right and the bottom of the matrix, and the corresponding edges in equality subgraph are shown in bold font. The equality subgraph is shown in Figure 9.3c, it doesn't have a perfect matching. So the labels are changed in Figure 9.3d, and a perfect matching, which is also the optimal matching, is found in Figure 9.3e.

The solution of the optimal assignment problem is also referred to as Maximum Weight Bipartite Matching (MWB).

9.3 MWB Matching on ICA components

The two vertex sets can taken to represent features from two datasets, and a edge in the matching indicates the two features at two ends have correspondence. Usually, the edge is weighted, and the weight represents the similarity between two vertices.

Given a number of feature components for each dataset, the similarity between two datasets is derived from the similarity of their feature components, which will be described in next paragraph. We do not have prior knowledge on how these components correspond to each other, so we test each component in one dataset against every component in the other dataset (the similarity measure between two components is addressed in next subsection). Now we have a complete weighted bipartite graph, of which each edge has a weight representing the similarity score between these two components. We then find the MWB matching in this bipartite graph, and use the sum of the weights as the similarity between the two datasets.

In this work, similarity between any two individual components is handled as if they are two activation maps. We first threshold the 1% top voxels, then apply the matching algorithms discussed in the earlier sections. A diagram of the process of matching ICA components is shown in Figure 9.4.

9.4 Notes

This chapter includes a part of [Bai et al., 2007c], which is co-authored by Bing Bai, Paul Kantor, Ali Shokoufandeh, and Deborah Silver.

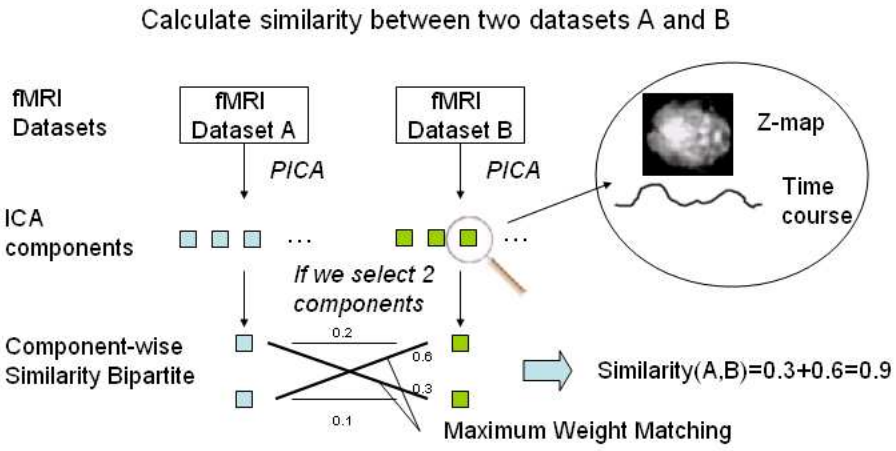


Figure 9.4: The process to calculate similarity between datasets.

Chapter 10
Data and Testing framework

10.1 Data Processing and Methods

10.1.1 Term definitions

In fMRI, there is considerable variation in the use of terminology and we briefly define the use of some key terms in this note. Note that these definitions may not be consistent with those in all other papers.

1. *Run*: A 3D brain image series that is scanned during one experimental session, with one subject continuously under observation.

2. *Condition*: A chosen stimulus. A typical run will contain at least two different conditions, one of which may be designed as a control.

3. *Experiment*: An fMRI study undertaken for cognitive research. An experiment will generally have many runs, with multiple subjects.

4. *Dataset*: For GLM-based methods, a dataset is a t-map. For ICA-based methods, a dataset is the same as "Run". (including control).

10.2 Testing database

In this preliminary research, we use data from 5 different experiments. The list of experiments and brief descriptions are shown in Table 10.1. The conditions of these experiments are shown in Table 10.2. In some of the experiments, namely,

Table 10.1: Experiments

Exp	Description	Publication
Oddball	Recognition of an out of place image or sound	
Event perception	Watching either a cartoon movie or real film of a human being	[Zaimi et al., 2004]
Morality	Subjects make decisions about problem situations having or lacking combinations of moral and emotional content	[Greene et al., 2001]
Recall	Study and recall or recognition of faces, objects and locations	[Polyn et al., 2004]
Romantic	People in love see pictures of their important others, or of non-significant people	[Aron et al., 2005]

"Event perception" and "Oddball", different conditions are performed only in separate runs. For the rest, several conditions will occur in each run.

10.3 Preprocessing and Postprocessing

The raw fMRI images from different scanners, or for different subjects are not comparable. Raw signals contain many kinds of noise. Thus fMRI images go through the following processing steps, via fMRI processing package FSL [Smith et al., 2001], before either GLM or ICA is applied:

1. Apply motion correction to align all 3d images in a time sequence to the same position, to correct the effects caused by small movements of the subject during the experimental run. Motion correction is a registration procedure to map all volumes in an fMRI scan to one certain volume (reference volume), with a rigid-body transform. In our experiments, the reference volume is the one in the middle of the sequence. The operations used in transform are 3 translations and 3 rotations, total 6 degrees of freedom (DOF).

Table 10.2: Datasets and Numbers

Experiment	**Condition**	TR(s)	Volumes	**# of datasets**
Oddball	Auditory	2.0	150	4
	Visual	2.0	150	4
Event Perception	House Active	1.5	110	28
	Study Active	1.5	210	25
Recall	Study Face	1.8	510	27
	Study Object	1.8	510	27
	Study Location	1.8	510	27
	Try to think of Face	1.8	510	27
	Try to think of Object	1.8	510	27
	Try to think of Location	1.8	510	27
	Recall Face	1.8	103	9
	Recall Obj	1.8	103	9
	Recall Loc	1.8	103	9
Morality	M+E+	2.0	150	50
	M+E-	2.0	150	50
	M-e-	2.0	150	50
Romantic	Neutral Face	5.0	144	15
	Positive Face	5.0	144	15
Total(N)				430

2. Remove the skull since it is not a part of a brain.

3. Apply spatial smoothing filters to control spatial noise. The filter used here is a 5mm Full Width at Half Maximum (FWHM) Gaussian filter, which is the default value in FSL.

4. Apply a temporal high pass filter to remove low frequency trends over time due to increasing temperature of the device or other factors. The algorithm used here is a described in [Woolrich et al., 2001]: "This approach fits and removes Gaussian-weighted running lines of fixed width using a least square fit...". The fixed width we are using are 50 scans (not seconds).

Another important issue is registration. Different brains have the same structure, but their shapes and sizes may be slightly different. Registration is the operation that transforms brain images onto a standard brain template to allow

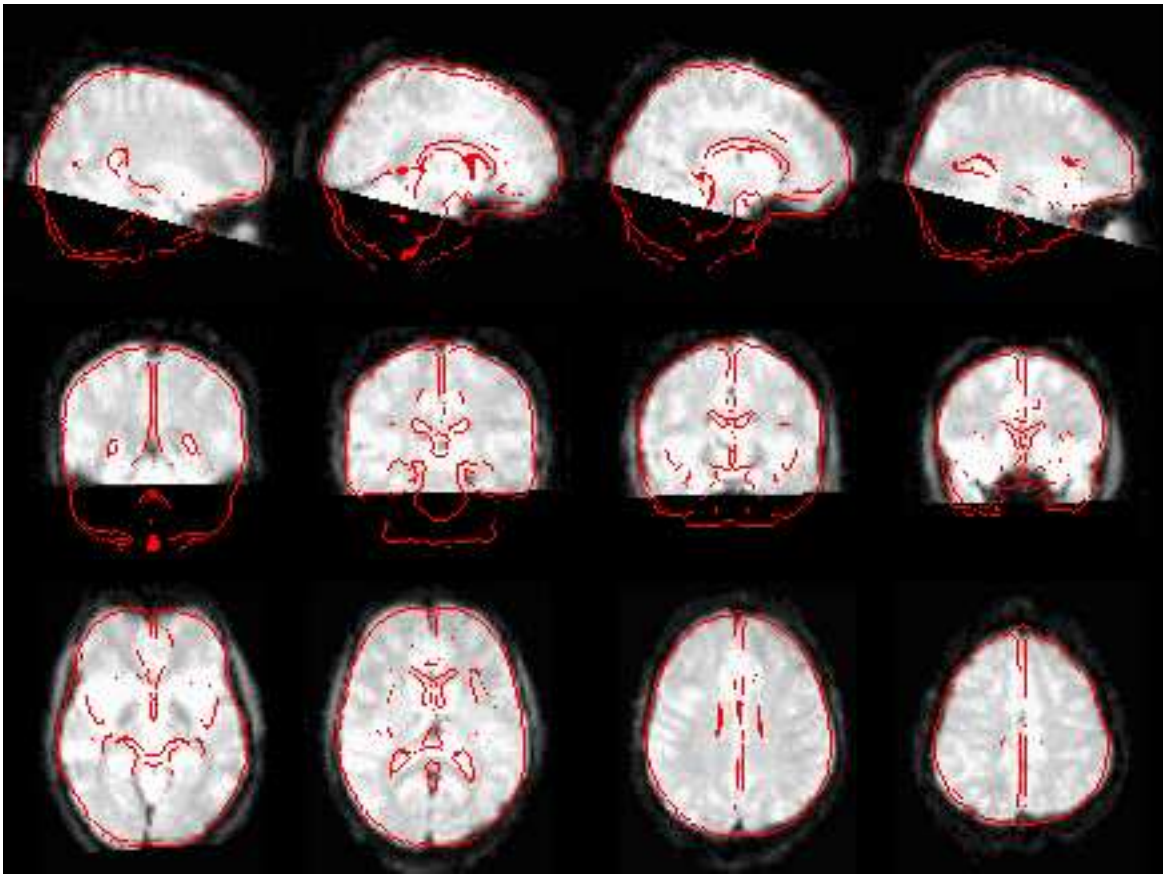


Figure 10.1: A registration of a sample fMRI image. 4 sample slices (gray scale) are shown in each of 3 views. The red line contours are the corresponding slices in the standard brain template.

inter-subject comparisons. Note that registration is not necessarily a preprocessing step. In this work, we choose to do registration after the activation maps (or ICA components) are generated. The "standard space" has smaller voxels, and thus building activation maps would take longer in the standard space. However, the apparently larger number of voxels does not really represent more information, since they are calculated from the smaller number of voxels in the raw image. Thus there is no loss in doing voxel selection before the transformation. The registration we apply here is the affine transform with full 12 DOF (translation, rotation, scaling, shearing). Figure 10.1 shows an example of the registration.

Before we extract feature voxels from t-maps in standard space, we need to mask every t-map with the common part of the brains scanned in all fMRI experiments in the testing database. Different portions of brains are scanned in different experiments, due to the decisions made by experimenters. Taking the common regions in fMRI scans prevents the artificial boost in performance because the images from the same experiment being similar only because they share a part of the brain which has not been scanned in other experiments.

10.4 Evaluation scheme

Our testing scheme is built on an information retrieval framework. We use every image as a query, and evaluate the performance of our method by checking the returned ranked lists. A retrieved image is considered "relevant" to the query only if they are both for the same experimental condition.

As noted, different experiments have different numbers of datasets. In this case, average precision will not behave as one might like, and we are suspicious of using it. Instead, we use the "area under the ROC" [Mason and Graham, 1982] (for the sake of simplicity, we call this "ROC area") to evaluate a retrieval method. For a retrieved list, suppose the number of relevant elements is m and the number of non-relevant elements is n. The ROC curve starts at the origin $(0, 0)$. We traverse the ranked list from the top. If an element is relevant, the ROC curve goes up by a step $1/m$; otherwise, ROC curve goes to the right by a step $1/n$. If the area under the ROC is 0.5, then the retrieval method is no better than random selection. Generally, the retrieval performance is considered good if the area under the ROC (AUC) is greater than 0.8. Figure 10.2 shows an example of ROC curve.

We use each of the datasets as a query, and rank all other images by their similarity with the query. We then have a AUC for every image. We can plot the

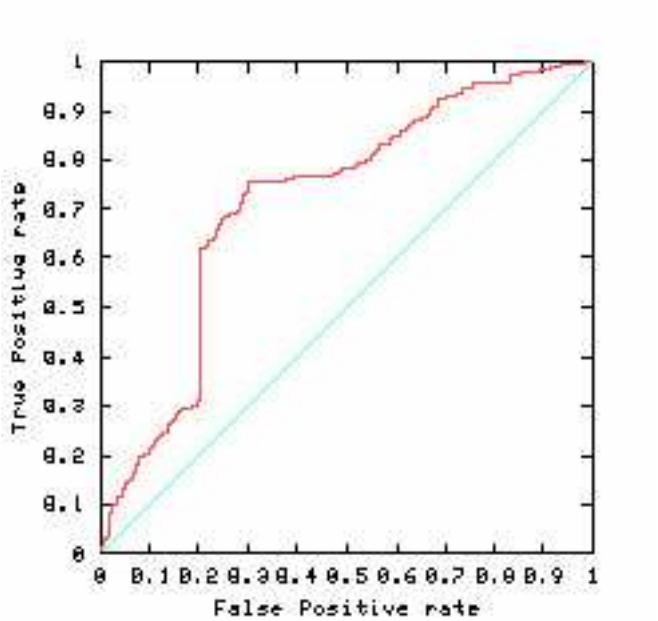


Figure 10.2: ROC curve.

histogram of these AUCs, as shown in Figure 10.3. We simplify the discription of this distribution by looking at only the mean value and the standard deviation.

10.5 Notes

This chapter includes parts of [Bai et al., 2006, 2007a], which are co-authored by Bing Bai, Paul Kantor, Nicu Cornea, Ali Shokoufandeh and Deborah Silver.

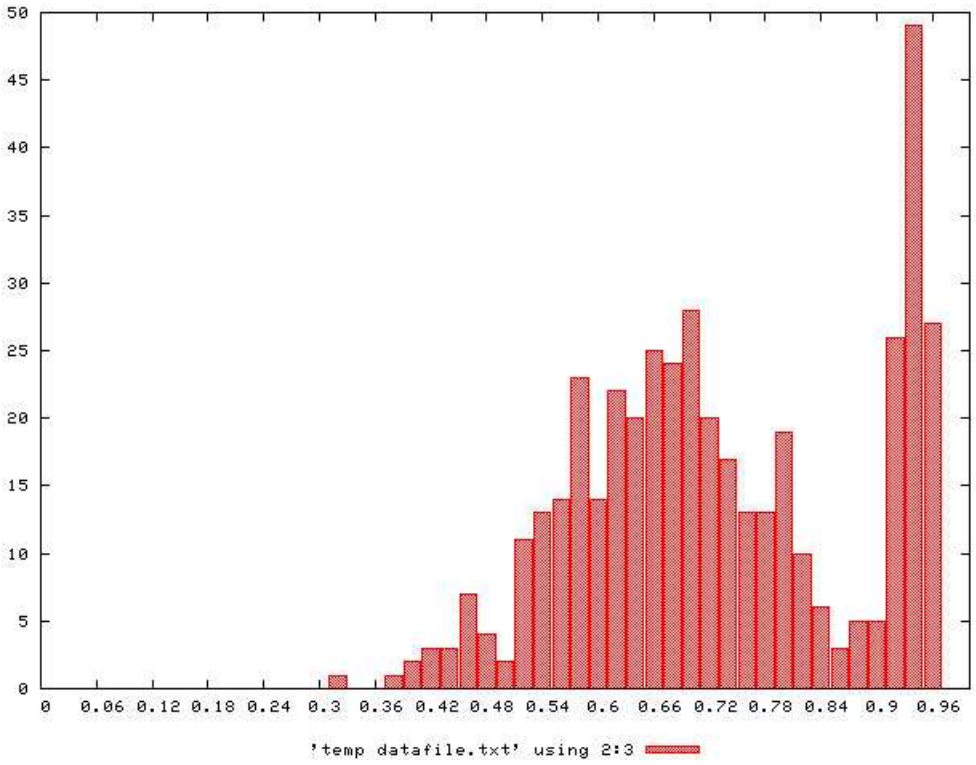


Figure 10.3: A sample histogram of AUC, generated by simple overlap on 1% voxels selected from FSL t-maps.

Chapter 11
Compare Feature Selection Models in GLM

11.1 Validating the Single Peak Non-Negative (SPNN) model

11.1.1 Validation tests on synthetic data

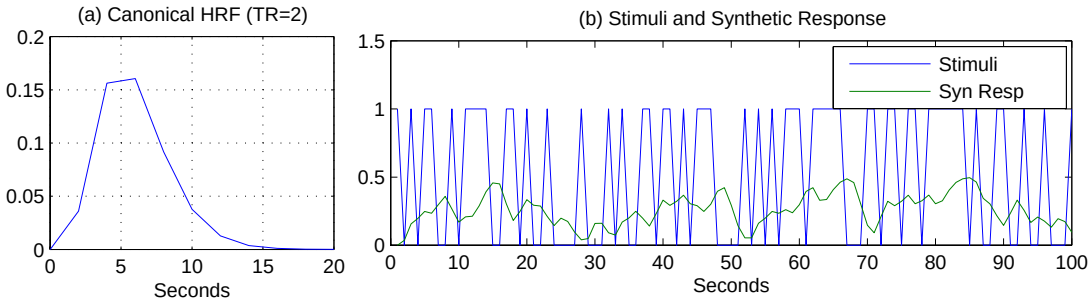


Figure 11.1: Synthetic data

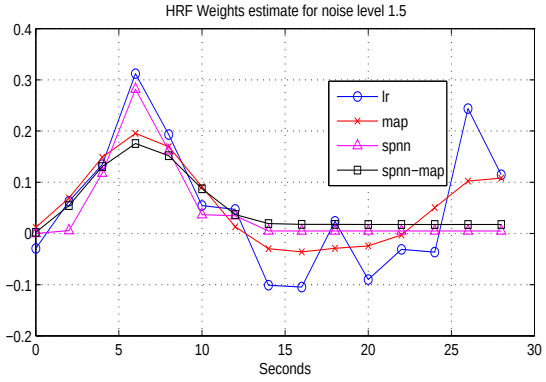


Figure 11.2: Recovered weights from synthetic data

In this section, we generate a synthetic brain response by convolving a hypothesized HRF with a hypothesized stimulus. A good model should recover the

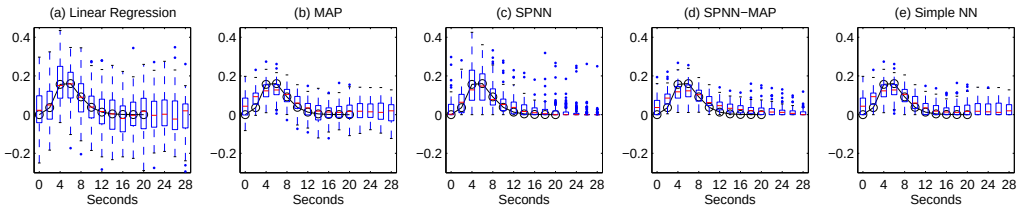
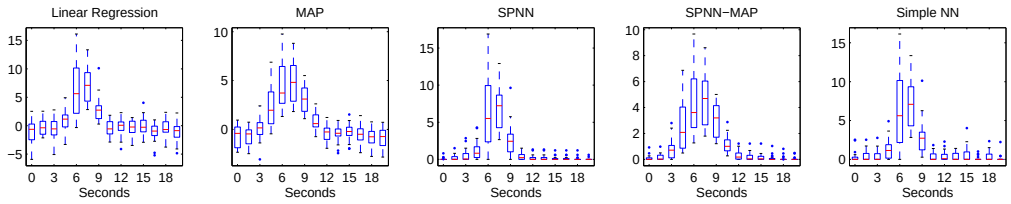


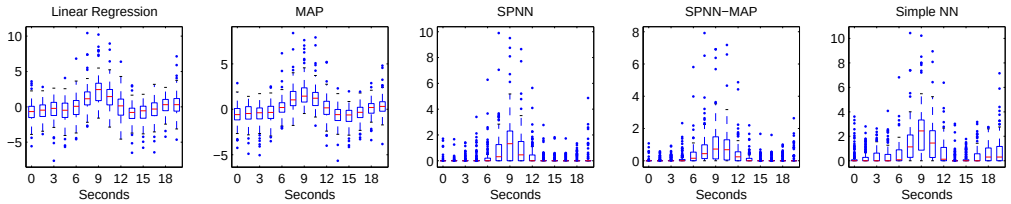
Figure 11.3: Box plot for weight estimation for 100 repeated experiments

HRF from this synthetic response, under a reasonable amount of noise. We select the HRF as a single gamma function with the sampling period (corresponding to TR) set to 2 seconds (shown in Figure 11.1 (a)), because it is a typical setting in many fMRI experiments. We take the time range from 0 to 20 seconds, requiring 11 weights in all (not including the constant w_0). The stimulus is a random binary sequence of length 100, with equal probabilities for "0" and "1". They are shown, together with generated synthetic response in Figure 11.1(b). We then add noise to this explanatory variable to simulate an "observation" time series. The noise follows a normal distribution $N(0, \sigma^2)$. We call σ^2 the *Noise Level*. When the noise level is set to 0, all methods give very good estimates. We set the noise level to 1.5, which is fairly large compared to the synthetic response in Figure 11.1.

We compare the recovered FIR HRF given by, linear regression (LR), MAP, SPNN and MAP combined with SPNN constraints (SPNN-MAP). We set the hyper parameters of MAP to $h = .3, v = .1, var = 1$. Note that we are not trying to find optimal values for these parameters. The choice of these hyper parameters will not affect our conclusion, as long as the same values are used in all experiments. To test the robustness of the methods when the width of the HRF is unknown, we set $n = 15$ instead of the real length of the HRF $n = 11$. We see that pure linear regression (LR) is seriously overfitting, with many changes of sign. Although MAP gives a smooth curve and a better fit, it still produces a large negative dip, and a heavy bump at late times. SPNN and SPNN-MAP both achieve good performance, but the curves of SPNN-MAP are smoother.



(a) Oddball auditory, 4 datasets, 20 voxels



(b) Event perception [Zaimi et al., 2004], 25 datasets, 125 voxels

Figure 11.4: Parameter estimation for real fMRI datasets

11.1.2 Validating tests on real data

As examples, we show experimental results from a few experiments in depth, but the results in other datasets are quite similar. In Figure 11.4, we compare the 4 methods on the following experiments. location that has been shown earlier to them. We select time series for the top 5 voxels from each dataset, which have the largest t-values in GLM analysis with double gamma function. This is to avoid selecting inactive voxels. As we observed for synthetic data, the SPNN method reduces the variance in HRF tail, while MAP reduces the variance overall. Their combination gives the most robust parameter estimation.

11.1.3 Discussion

The move from phantom data to real data has provided some expected confirmations, and some puzzling new phenomena. As expected, the SPNN-MAP performs quite well, producing smooth parameter estimates, with relatively tight variation, and with very few "late lag" artifacts. On the other hand, the precise location of the peak in all of the models is surprising. Assuming that we can read

the "typical shape" of a solution by the shape of the median points, we see that all methods, whether or not they are constrained to yield positive weights, seem to peak considerably later than the very model that was used in the analysis of the data (recall that we selected these voxels by GLM based on double gamma function). There are several possible explanations, which invite further study. One possibility is that the selected voxels (which are in real brains) may have a response which matches "well enough" to the double gamma function, but is even more accurately described 'by a single peak which occurs somewhat later than is usually believed.

While these questions remain open, the results shown here indicate that with modern computation methods the apparent complexities of quadratic programming can be surmounted, and that the extensions to the FIR approach show promise in the robust analysis of realistic and noisy data.

We repeat the experiments 100 times, with random noise generated with the same noise level of 1.5. We show boxplots [Tukey, 1977] for each method in Figure 11.3. The bottom of a box is the value of the first quartile, the top is that of the third quartile, while the bar in the middle is the median. There are also two "nails" (called "whiskers") that stick out upward and downward. They indicate the maximum and minimum values, or 1.5 times the height of the box (which is called inter-quartile range (IQR)) from the median, which ever is closer to the median. For the latter case, all values outside that range of whiskers are considered "outliers" (shown as dots). We see the medians in all 4 plots match very well to the true HRF, which is marked with small circles. Due to the smoothing factor, the estimated curves for MAP and SPNN-MAP are flattened a small amount. Linear regression apparently has largest variance in parameter estimation. MAP effectively reduces this variance. However, MAP shares the weakness of linear regression that the variance does not diminish as the lag increases, which is manifested as the oscillation in Figure 11.2. On the

other hand, the estimates of SPNN decrease properly for larger lag, but it has a relatively large number of outliers. Finally, in Figure 11.2, the SPNN-MAP method combines the advantages of both SPNN and MAP. In Figure 11.3, we still see a number of outliers in the figure for SPNN-MAP, but note they are actually below the upper whiskers of MAP estimations. SPNN-MAP makes the IQR quite small, and points are more likely to be considered as "outliers". Simple NN (non-negativity) serves as a null model. It is the same as MAP except all the negative values are set to 0. we see the variation does not decrease at late times as much as for SPNN and SPNN-MAP. This method is included to show the performance gain produced by SPNN is not trivial.

11.2 Convergence of the multiple regression method for FIR model

We have done the following experiment on synthetic data:

1. randomly generate (in uniform distribution) an observation time series $y(t) \in (0,1), t \in [1,N]$.

2. randomly generate (in uniform distribution) c stimulus time series $S^i(t) \in (0,1), i \in [1,c], t \in [1,N]$.

3. randomly generate (in uniform distribution) initial points $\mathbf{a}^0, \mathbf{w}^0$, where $a^0(i) \in (-100,100), w^0(j) \in (-1,1), i \in [1,c], j \in [1,n]$ are randomly generated in uniform distribution.

4. For a dataset generated with certain values of c, n, N, we run the experiment r times, and count the number of distinct minima. Two points $\mathbf{a}, \mathbf{a}'$ are considered different if $(\mathbf{a}-\mathbf{a}')(\mathbf{a}-\mathbf{a}') < 0.0000001$.

Table 11.2 shows the distribution of number of minima over different values of c, n, N. Note that if two vectors of a differ only in signs, we consider them as

Table 11.1: Number of local minima for different values of c, n, N

(a) N=30, r=20

	n=5	n=6	n=7	n=8	n=9	n=10
c=1	1	1	1	1	1	1
c=2	**2**	1	**2**	**2**	1	1
c=3	1	1	**2**	1	1	**3**
c=4	**2**	**2**	**3**	**2**	**2**	**2**
c=5	**3**	**2**	**3**	**2**	1	**3**

(b) N=130, r=20

	n=5	n=6	n=7	n=8	n=9	n=10
c=1	1	1	1	1	1	1
c=2	1	1	1	1	1	1
c=3	1	1	1	**3**	1	1
c=4	1	1	1	**3**	1	1
c=5	**3**	**2**	1	**2**	1	**3**

(c) N=230, r=20

	n=5	n=6	n=7	n=8	n=9	n=10
c=1	1	1	1	1	1	1
c=2	1	1	1	1	1	**2**
c=3	1	1	1	1	1	1
c=4	1	1	1	1	1	1
c=5	1	1	1	1	1	**2**

one. The numbers with bold face indicate more than one minima. Although we are not completely clear about the exact pattern of convergence, we found that the number of minima decreases as the number of observations N increase and increases as the length of $\mathbf{a}, \mathbf{w}$.

The last concern about this algorithm is its speed. We have not generated the theoretical time bound yet. We conduct the following experiment to get an empirical estimate.

1. Let $c = 5, n = 10, N = 230$, which is a typical fMRI setting.

2. Randomly generate (in uniform distribution) 10 observation time series $y(t) \in (0, 1), t \in [1, N]$ and 10 stimulus sets $S^i(t) \in (0, 1), i \in [1, c], t \in [1, N]$.

3. Randomly generate initial points $\mathbf{a}^0, \mathbf{w}^0$, where $a^0(i) \in (-100, 100), w^0(j) \in (-1, 1), i \in [1, c], j \in [1, n]$ in uniform distribution. Repeat this 20 times and calculate the average number of iterations $m_i, i \in [1, 10]$ before the algorithms converges with $|\mathbf{a}^{t_c} - \mathbf{a}^{t_c-1}|^2 < 0.0000001$ for each of the 10 observation/stimulus sets generated in step 2.

4. Calculate the mean iteration number and its standard error of $m_i, i \in [1, 10]$.

The mean iteration number is **26.7** with standard error **5.8**.

11.3 Initial point selection for LLFOM

In our pilot studies, we have done some searching for a good initial points. We tried some initial points and found that $A = 0.1, B = 0.1, C = 0.2$ always converges to a visually satisfying results. Note that we are performing grid search for optimal delay τ. Based on prior knowledge of the hemodynamic response, we limit the τ to the interval $[0, 10]$ seconds.

Since we have a large number of non-linear programming problem to solve (for each voxel we have $\lceil 10s/TR \rceil$ problems), we can not afford to try many initial points in searching. So in our experiments we used only one initial point $A = 0.1, B = 0.1, C = 0.2$.

In Figure 11.5, we show an example of norm-2 fitting for the non-linear model.

11.4 Retrieval effectiveness

In this section, we compare the performance of different feature selection models in the framework of general linear model. We use only one matching algorithm in this section: the simply overlap (Jaccard distance). In next section we will

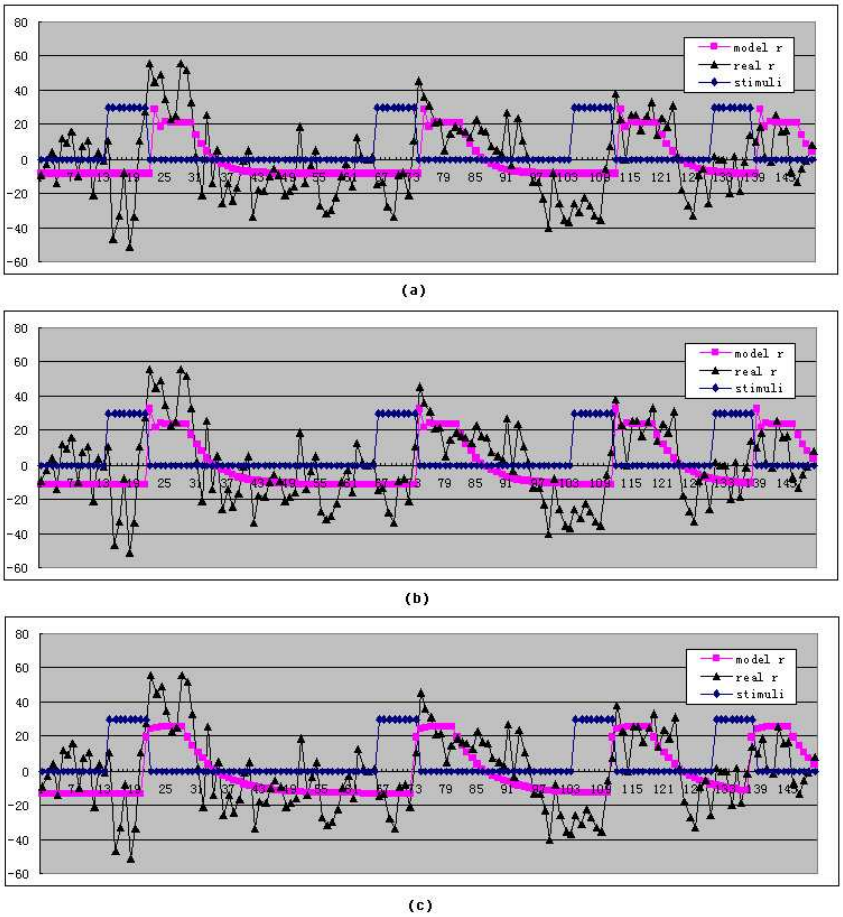


Figure 11.5: LLFOM model fitting for one voxel of the Morality dataset. τ is set to 8, 7 and 6, corresponding to 16 seconds, 14 seconds, and 12 seconds, respectively.

Table 11.2: The length of FIR filter for experiments

Experiments	TR	length of FIR
Oddball	2.0	15
Event perception	1.5	20
Morality	2.0	15
Recall	1.8	17
Romantic	5.0	6

Table 11.3: Average AUC (430 datasets) for different feature selection methods (Mean/Standard Error of the mean).

Canonical Multiple	.662	.007
Canonical Single	.677	.007
MAP Multiple	**.719**	.006
MAP Single	**.715**	.007
MAP SPNN Single	.672	.007
SPNN Single	.652	.008
LLFOM	.650	.007

see that The Jaccard distance gives fairly good results, and it is very simple to perform.

For FIR models, we set the length of the finite response to be about 30 seconds. That is, the length of FIR filter is listed as shown in table 11.2.

Table 11.3 shows the AUCs of all the methods we tested. "CAN", "MAP", "SIN" and "Mul" stand for "Canonical HRF", "MAP HRF", "Single regression" and "Multiple regression", respectively. The other models are "SPNN with single regression", "MAP SPNN with single regression" and "LLFOM".

We have the following observation for above table 11.3.

1. The MAP FIR model performs better than all other models in retrieval. We can see that the differences are highly significant, for both single-variate and multi-variate approaches.

2. The multiple regression does not provide better retrieval performance than

single regression. This is confirmed by both "MAP FIR" model and "Canonical HRF" model. Figure 11.6 compares the four methods for separate conditions (see Table 10.2). Each method is better for *some* conditions. We shall return to this in details in the discussion.

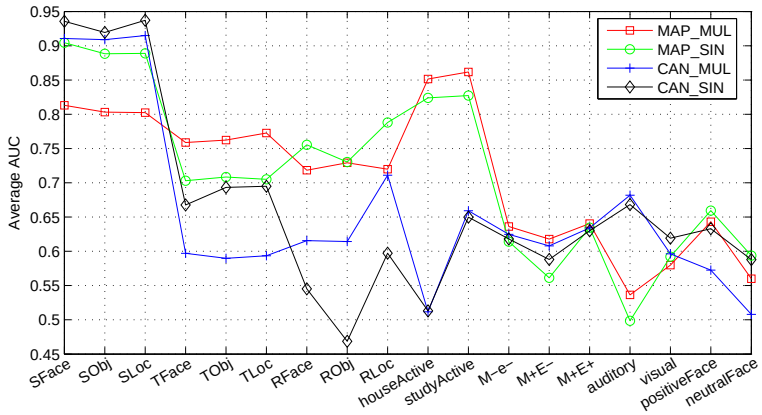


Figure 11.6: Average of area under ROC curve for 4 methods.

3. Despite its "visual advantage" as presented in section 11.1, the SPNN model does not give the performance improvement we were expecting, whether it is combined with MAP or not. Actually, if we look back at the figure 11.4, we see both simple FIR model and MAP FIR model gave negative weights, probably corresponding the "initial dip" and "undershooting". Maybe the non-negativeness is a too strong restriction. This leads us to the "Peak-valley" FIR model presented in future work (14.1).

4. The nonlinear model LLFOM is not so competitive in this comparison. We think a possible reason is that we only start from one initial point in the search, partially because searching from many starting points takes too much time. We are looking for ways to speed up this searching process (probably by avoiding the naive grid search), so we can afford to conduct more complete search.

11.5 Discussion

11.5.1 Multiple Regression vs. Single Regression

The FIR model, with MAP smoothing, which seems to be a more realistic way to describe the variations, across the brain, in the anatomy supplying blood, *does* also yield significantly better performance in the retrieval setting. This suggests that it may be worth the added effort to use smoothed FIR analysis when preparing data for retrieval across different experiments, and different laboratories.

On the other hand, the expected superiority of using multiple independent regressors to select the voxels characteristic of several cognitive conditions occurring in the same run, is not confirmed. This lead us to a more detailed examination of *why* it was expected to be better, and a new hypothesis.

Our argument in favor of using multivariate regression relied on the assumption that an individual voxel may be activated by several conditions, all occurring in the same experimental run. Using all but the condition of interest as a contrast has the effect of making the estimates of correlation with the signal less accurate. This makes the t-value smaller, and makes the voxel less likely to be selected as a feature. On the other hand, conditions that activate same voxels are harder to tell within the same run. These two contradictory factors may dominate in different experiments. As shown in Figure 11.6, for some types of experiments the multivariate regression *is* (e.g., M+E+, M+E- and M-e-) more effective, while for some of them (e.g. SFace, SLoc and SObj) it is not.

This relationship will be further investigated in future work.

11.6 Notes

The results we present in this section are generated by straightforward GLM modeling after simple preprocessing steps presented in section 10.3. However,

in main stream software package such as FSL, more advanced features such as temporal autocorrelation corrections, structure preserving spatial smoothing are applied along with GLM. The results from these software packages are usually better than our results. We plan to apply these techniques in our FIR models in the future.

This chapter includes parts of [Bai and Kantor, 2007, Bai et al., 2007b], which are co-authored by Bing Bai, Paul Kantor, and Ali Shokoufandeh.

Chapter 12
Results of IR matching algorithms

In this chapter we compare matching algorithms described in chapter 8. The t-maps used here are generated by 3 methods: 1) Canonical HRF with single regression, referred to as "CAN SIN". 2) Canonical HRF with multiple regression, adding autocorrelation corrections and structure preserving spatial smoothing. This result is generated with FSL, thus is referred to as "CAN FSL". 3) MAP FIR with single regression, referred to as "MAP SIN". Note that we are not comparing the feature selection methods in this section, but only the matching algorithms. Thus the comparisons are examined within each individual feature sets.

In Table 12.1 we list the average ROC area for different similarity measures. "Cosine", "Euclidean", "Manhattan" are 3 naive methods which we consider as "baseline". "PCA whole t-map" and "LSI whole t-map" are two dimension reduction technique. The pair-wise similarity measure is Cosine on the first 10 components. The rest of the methods uses the 1% voxels of the voxels with largest t-values or absolute t-values. There are two things worth mentioning. 1) The methods based on "whole t-maps" do not apply to the "FIR" model since the signs of the HRF and the weights could be either both positive or both negative, depending on the initial point from which the search starts. In this case, the distance measures sensitive to signs (including "Euclidean", "Manhattan" and eigenvalue based methods such as PCA and LSI) do not make sense. 2) The Mahalanobis is only applied on dimension reduced coordinates of LSI, because the covariance matrix for raw voxels features will be too expensive to calculate.

Table 12.1: Average AUC for 430 datasets. The left column is for the t-maps built from GLM with canonical HRF, single regression. The middle column is for the t-maps built from GLM with canonical HRF, also with autocorrelation correction and structural preserving spatial smoothing performed by FSL. The right column is the t-maps built from MAP FIR, single regression. The boldface numbers are two best scores.

Method	**CAN SIN**	**CAN FSL**	**MAP SIN**
Cosine	$.646 \pm .007$	$.704 \pm .007$	
Euclidean	$.573 \pm .006$	$.528 \pm .011$	
Manhattan distance	$.570 \pm .006$	$.530 \pm .011$	
PCA whole t-map	$.699 \pm .007$	$\mathbf{.791 \pm .006}$	
LSI whole t-map	$.678 \pm .006$	$.688 \pm .007$	
Overlap (Fuzziness Radius 0)	$.677 \pm .007$	$.733 \pm .007$	$.715 \pm .007$
Overlap (Fuzziness Radius 1)	$\mathbf{.717 \pm .007}$	$.761 \pm .007$	$\mathbf{.737 \pm .007}$
Overlap (Fuzziness Radius 2)	$\mathbf{.721 \pm .007}$	$\mathbf{.772 \pm .007}$	$\mathbf{.728 \pm .007}$
TFIDF	$.680 \pm .007$	$.735 \pm .007$	$.657 \pm .008$
LSI 1% voxels (10 comps)	$.665 \pm .006$	$.755 \pm .006$	$.689 \pm .007$
LSI 1% voxels Mahalanobis (10 comps)	$.640 \pm .006$	$.552 \pm .007$	$.632 \pm .010$

We see that the overlap methods based on thresholded "t-maps" gives more stable performance than other methods. Although we do not list the results for all the radiuses here, but the observation is that the best performance is reached usually with radius 1 or 2. This is qualitatively as we expected. An appropriate amount of fuzziness can correct for some registration error, and for anatomical variations among individual brains, while too much fuzziness eliminates the difference between the activated and non-activated regions. This also provides additional support for the choice of thresholded t-maps. As the threshold is lowered, we are more likely to be including random noise, which will hurt the quality of retrieval, rather than improving it. More importantly, this thresholding approach makes inverted indexing quite efficient.

The number of components selected in dimension reduction techniques (LSI and PCA) are chosen to give the best average AUCs. Although the dimension reduction makes the retrieval performance certainly better than baseline methods, and one number in the table is very impressive, it is not as stable as fuzzy overlap.

12.1 The LSI components in brain images

In section 10.1, we motivated LSI simply by an analogy between textual retrieval and brain image retrieval. Nevertheless, we suspect that LSI has important anatomical and cognitive meanings that might lead to better ways to index.

First, it is quite possible that a cognitive neural process is decomposed into sub-tasks by the brain. For example, many stimuli are delivered as visual signals. So the corresponding brain activations should include the basic visual reaction, recognition of the objects, and/or higher level cognitive processes. Second, oxygen is conveyed by vessels, and big vessels like arteries will be activated for many different stimuli. With LSI, we may hope to isolate these factors, and link some LSI basis vectors to more fundamental cognitive processes, while providing retrieval

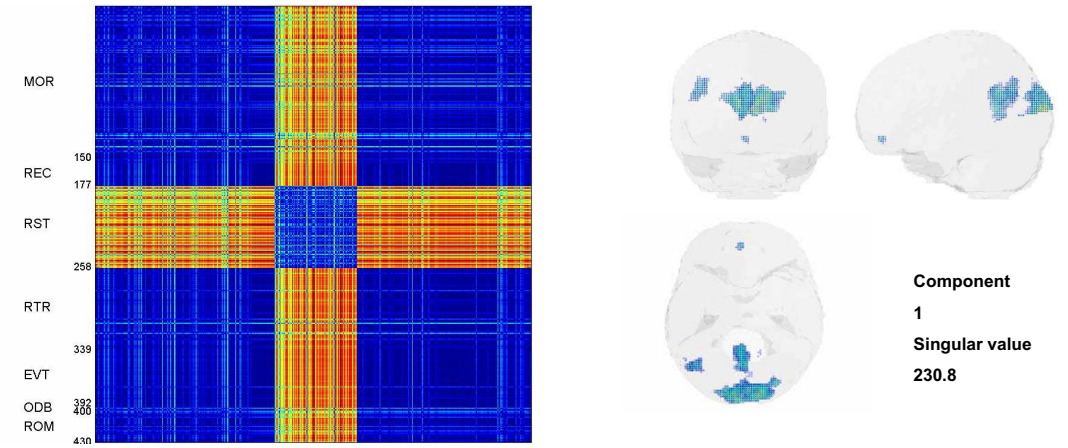


Figure 12.1: The 1st (U, S, V) triplet of the LSI with the whole brain t-maps.

for a wide range of queries.

From Figure 12.1 to Figure 12.7, we show the top 7 (U, S, V) triplets generated by LSI on whole brain t-maps from FSL. The square on the left shows the pair-wise distance between the weights of each components in all 430 datasets. For example, Figure 12.1 shows the first triplet, the position $(i, j) i \in \{1, 2, ..., 430\}, j \in \{1, 2, ..., 430\}$ shows the difference between $\|V_{i,1} - V_{j,1}\|$. A shorter wavelength of the color (blue) represents a smaller value, a longer wavelength of the color (red) represents a larger value. The brain map on the right shows the 1% of the voxels with largest absolute values in this component. For example, the colored regions in the brain map of Figure 12.1 are the voxels with largest absolute values in the first vector of U, blue-like voxels represent very negative values, the red-like voxels represent positive values.

Before we go further in this discussion, let us look at condition-level retrieval performance. Table 12.2 shows a comparison between LSI and Simple Overlap, on t-maps built by FSL.

We see that although the total average AUC is not very different for Simple Overlap and LSI, the condition-wise AUC is quite different. For example, the

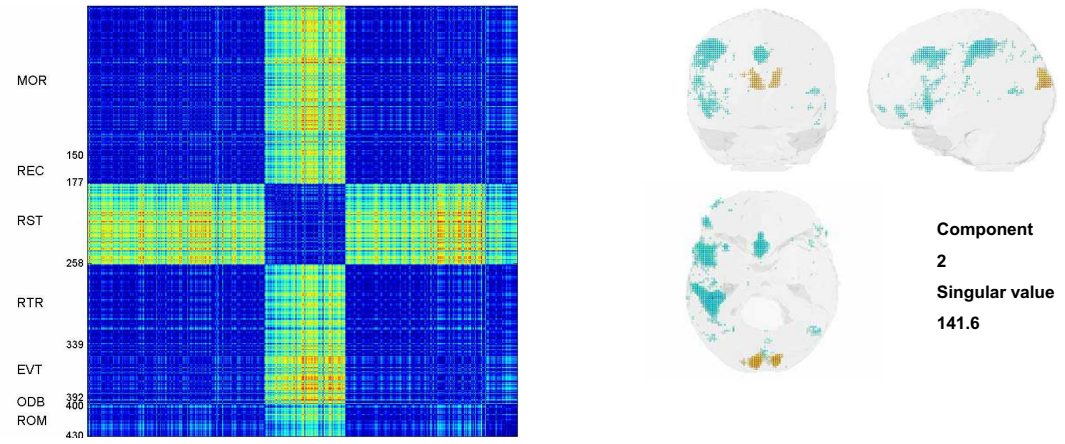


Figure 12.2: The 2nd (U, S, V) triplet of the LSI with the whole brain t-maps.

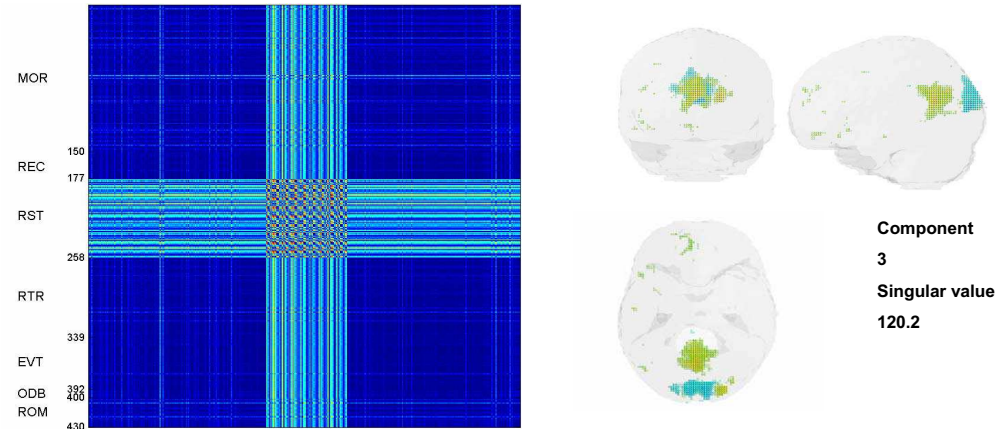


Figure 12.3: The 3rd (U, S, V) triplet of the LSI with the whole brain t-maps.

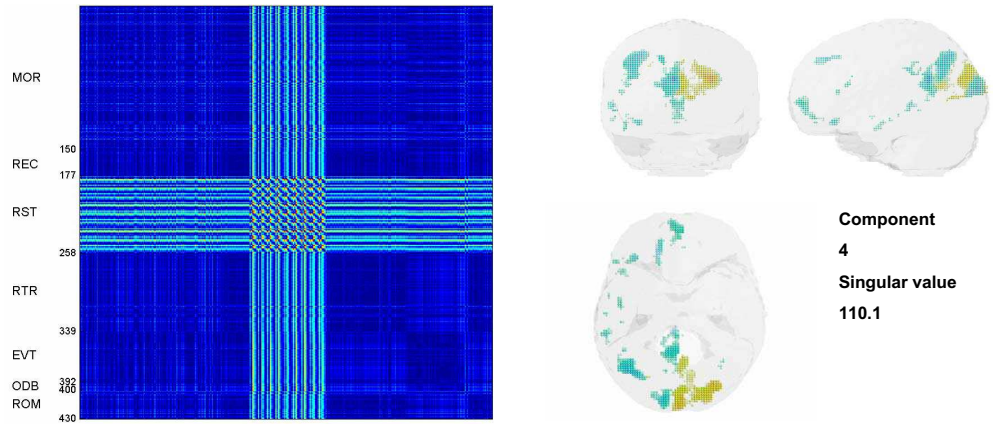


Figure 12.4: The 4th (U, S, V) triplet of the LSI with the whole brain t-maps.

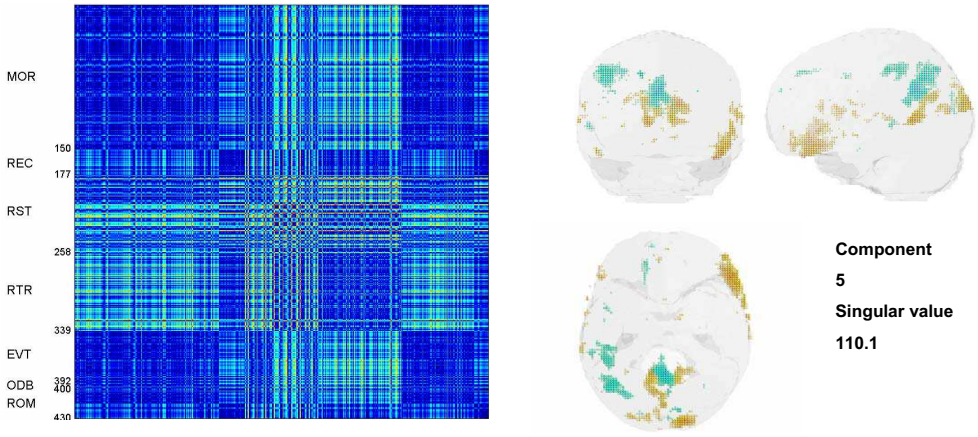


Figure 12.5: The 5th (U, S, V) triplet of the LSI with the whole brain t-maps.

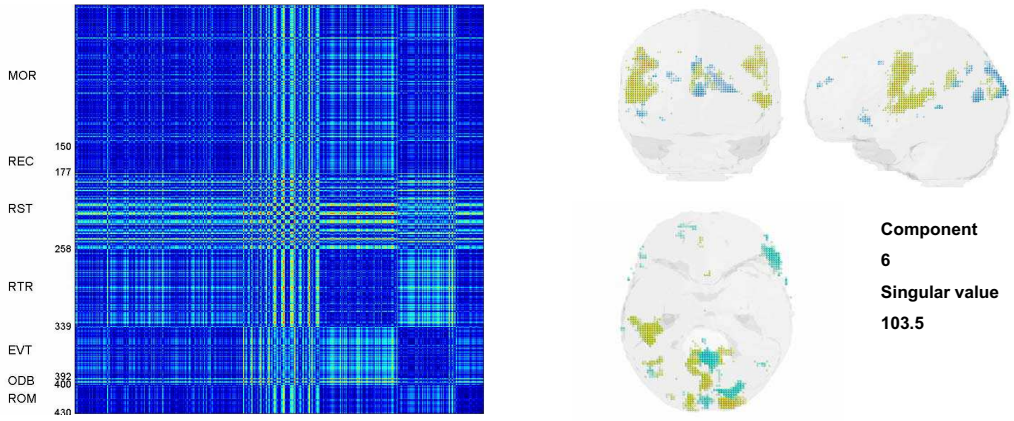


Figure 12.6: The 6th (U, S, V) triplet of the LSI with the whole brain t-maps.

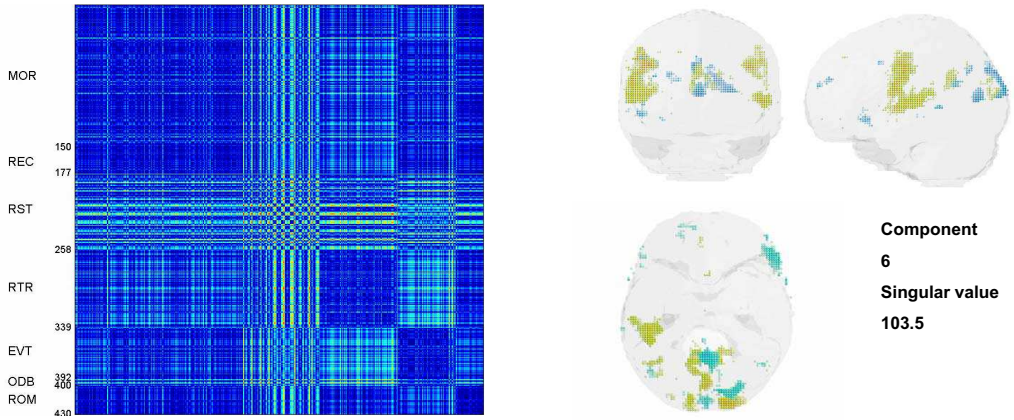


Figure 12.7: The 7th (U, S, V) triplet of the LSI with the whole brain t-maps.

Table 12.2: Condition level Average AUC comparison between LSI (whole tmaps) with Simple Overlap

Experiment	Condition	Simple Overlap	LSI
Oddball	Auditory	.499	.755
	Visual	.592	.706
Event Perception	House Active	.823	.864
	Study Active	.827	.774
Recall	Study Face	.904	.743
	Study Object	.888	.748
	Study Location	.889	.731
	Try to think of Face	.703	.522
	Try to think of Object	.708	.521
	Try to think of Locaction	.705	.542
	Recall Face	.755	.557
	Recall Obj	.730	.486
	Recall Loc	.788	.578
Morality	M+E+	.634	.772
	M+E-	.561	.679
	M-e-	.614	.718
Romantic	Neutral Face	.659	.611
	Positive Face	.594	.793

Simple Overlap method performs better in the "Recall" experiments, while the LSI is better in "Morality" and "Event Perception" experiments.

However, if we look at the first two LSI triplets, we can see that one of the "Recall" subsets "Study (Face, Object, Location)" is very clearly separated from other datasets. In other words, although there are components distinguishing some experiments, the cosine measure may hide this distinguishing ability. This motivates us to explore other similarity measures on LSI components.

12.2 Algorithm Efficiency

12.2.1 Space Cost

For Simple overlap, suppose the number of selected voxels in each dataset is S, the number of all voxels (in our case it is the intersection of all images) is W.

The inverted indexing method reduces the space by a factor $2S : W$. In our case, S/W is 0.01 since we take only 1% of the voxels. There is a factor of 2 because we need both forward and inverted indices. In a forward index, each element is a voxel ID. Similarly, each element in an inverted index is a image ID. Both of them can be represented by a 4-byte integers.

For the fuzzy overlap method, the number of indexed voxels will increase because neighbors are counted. If the neighbors did not overlap, the number of indexed voxels for fuzzy overlap with radius 2 would be 125 times (5^3) as many as that of simple overlap. In fact, activated voxels are often clustered, and thus many of neighbors actually overlap. In our experiment, as an empirical example, the space ratio of fuzzy overlap with radius 2 to simple overlap is only 11:1.

For TFIDF, in addition to the voxel IDs and document IDs, we also need to store t-values for each indexed element. This doubles the space because each t-value is a 4-byte floating point number.

For LSI, we do not use the inverted indexing. Instead, the SVD triplet (U, S, V) is stored. If the number of voxels is m, the number of datasets is n, and we keep the top t components, then the space needed is $m \times t + n \times t + t$. In our experiment t is only 10. Note that m is not the number of voxels in the whole image, it is the number of voxels that are activated in at least one image. In fact, these two numbers are very close in our experiment.

12.2.2 Time Cost

First, we consider the time cost of the quasi-cosine measures, including simple/fuzzy overlap and TFIDF. Let the number of selected voxels for each dataset be S, the number of images is n, and the number of voxels in the brain space is V. Then the expected length for each index entry is $L = NS/V$. When a query is submitted (with S voxels, of course), we need to traverse S lists in the inverted index to generate the ranked list. Each traversal takes $O(L)$ time. So in

the average case, one single retrieval takes time:

$$O(SL) = O(S^2N/V) \tag{12.1}$$

12.3 Notes

This chapter includes parts of [Bai et al., 2006, 2007a], which are co-authored by Bing Bai, Paul Kantor, Nicu Cornea, and Deborah Silver.

Chapter 13
Hypothesis-free retrieval methods based on ICA

In contrast to the methods of the previous chapter, we do not use any stimulus information in ICA-based methods. In this discussion we do not separate the multiple conditions applied in the same run. A retrieved image is considered "relevant" to the query only if they both correspond to the experimental conditions of the run.

13.1 Experiments and Alternative Approaches

In component selection, we select as the top 10 components those with the lowest expected time frequency (LFC). We compare its retrieval performance with 10 "highest expected frequency" (HFC) and 10 "randomly chosen" components (RDM). For image matching, besides Maximum Weight Bipartite matching (MWB), we also include a method in which the similarity between datasets is the maximum edge weight of the component similarity bipartite graph, and it is referred to as MAX in Section 13.2 (For example, in Figure 9.4, the MAX similarity between the two datasets is 0.6). All 6 component by matching combinations are tested.

13.2 Results

Table 13.1 shows the average area under the ROC. We can see that (1) the scores of MWB are always significantly better than MAX, (2) the score of "low

Table 13.1: Average ROC area for 360 datasets. 10 components are selected with LFC, HFC, and RDM, respectively. Two image similarity metrics, MAX and MWB, are included. Difference exceeding 1.96 standard errors are significant at 95% confidence. All pair wise differences among 6 methods are significant at 95% (Bonferroni corrected) confidence except the pairs marked a-a and b-b.

Method	MAX	MWB
Low Frequency	(a) $.666 \pm .005$	(b) $\mathbf{.729 \pm .005}$
High Frequency	$.645 \pm .006$	(a) $.665 \pm .006$
Random Components	$.592 \pm .004$	(b) $.716 \pm .006$
All Components		$.700 \pm .009$

frequency components with MWB" is the highest, and it is significantly better than all others.

Figure 13.1 shows the average ROC area for individual experiments. We can see the LFC with MWB has the most stable performance across all the experiments. Note that RDM with MWB gets a high score in Table 13.1, but this is only because this method does well for the experiment with largest size (Morality, 248 datasets), so the average is pulled up, although it performs poorly for other experiments.

All 6 methods based on ICA do rather well on the largest category. But the low frequency components, coupled with aggressive matching of each component to its best "partner" does the best, or very well on all categories of experiment. This is consistent with the facts noted earlier: (1) when we observe the HRF we expect low frequencies to be most informative and (2) since the ordering of the components is not logically fixed, pairing each component to its best match for another experimental run gives a more complete picture of the similarity.

13.3 Notes

This chapter includes parts of [Bai et al., 2007c], which are co-authored by Bing Bai, Paul Kantor, Nicu Cornea, and Deborah Silver.

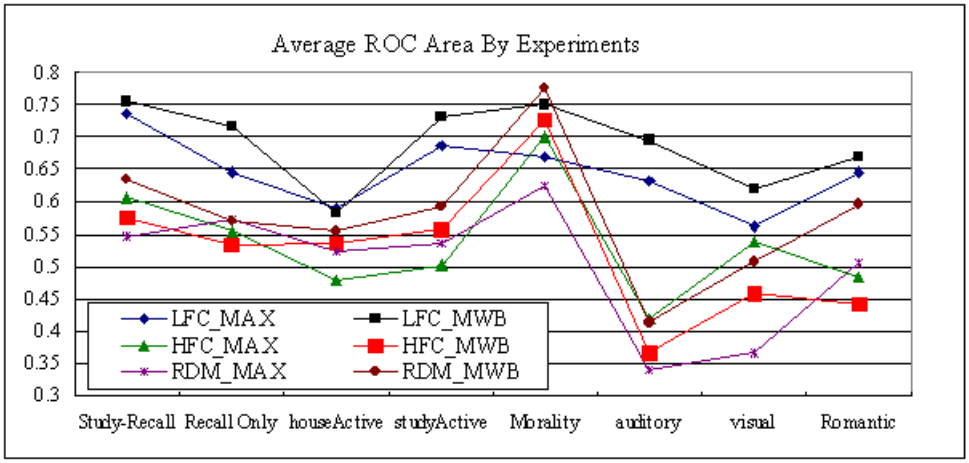


Figure 13.1: Average ROC area for individual experiments.

Chapter 14
Future work

14.1 Peak-Valley FIR model

The SPNN model can be seen as an example as the overfitting restriction-based on shape information. As the result indicated, the SPNN does not improve the retrieval performance significantly. If we look back at the results in Figure 11.4, we see that there is a considerable amount of negative weight that is lost in SPNN model. This motivates us to apply similar constraints in SPNN to model the negative undershoot. Specifically, we have

$$\begin{cases} w_{i-1} < w_i \quad , \quad i \in [2, p] \\ w_{i-1} > w_i \quad , \quad i \in [p+1, q] \\ w_{i-1} < w_i \quad , \quad i \in [q+1, n] \end{cases} \tag{14.1}$$

In these equalities, there is a peak p as before, and also a valley q to model the undershoot.

14.2 Inverted retrieval

Another interesting direction is what we may call "inverted retrieval". Instead of retrieving "similar images", we can retrieve "similar voxels". This is also potentially valuable in fMRI research. If we have already identified the functionality of a brain region, we might want to seek other regions with the same or similar function.

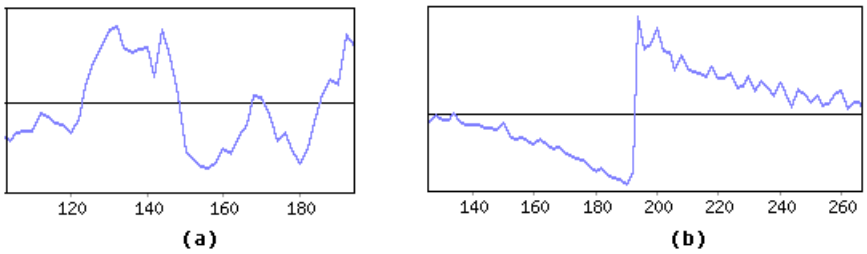


Figure 14.1: Two time courses with low frequency spectrum. (a) shows a hemodynamic response. (b) is usually considered to be a head motion artifacts. See [Calhoun et al., 2003].

In the IR framework, this task can be naturally implemented. For overlap/TFIDF approaches, we simply switch the roles of forward and inverted indices. For a given"query voxel", we rank the voxels in decreasing order of the number of images in which they were "activated" along with the query voxel. This same notion can also be extended to LSI by interchanging the "term/voxel matrix" U and the "document/image" matrix V. Quite generally, in retrieval algorithms like inverted indexing or LSI, "terms/voxels" and "documents/images" have a "dual" relationship, and an application in one direction can often find a meaningful counterpart in the other direction.

14.3 More heuristics on ICA components selection

In our experiments, the low frequency approach usually gives high rank to meaningful time courses, as shown in Figure 14.1 (a). However, in a few cases it also selects meaningless time courses. Figure 14.1 (b) shows one such example. It is usually referred to as an artifact caused by head motion (note we have performed motion correction, but the effect still appears sometimes). Prior knowledge of these patterns may be a valuable aid to low frequency component selection. Also, the number of components to select should be explored further. The 10 components used in this thesis is one that gives good results. However, using a fixed number for every experiment seems arbitrary and inflexible. We might improve

retrieval performance if the number of task-related components can be identified separately for each dataset, using frequency and/or other heuristics.

14.4 Note

This chapter includes parts of [Bai et al., 2006, 2007a,c], which are co-authored by Bing Bai, Paul Kantor, Nicu Cornea, Ali Shokoufandeh and Deborah Silver.

Chapter 15
Conclusions

Our goal is to build "content-based" fMRI retrieval systems, to support efficient data sharing schemes for a fast-growing functional MRI imaging corpus. The "content" in this context refers to specific cognitive processes, which are reflected as different activated brain regions. An effective retrieval scheme would be useful in neuroscience study, and, we hope in clinical diagnoses in the future.

Our work can be summarized as 2 categories – "hypothesis-based" methods and "hypothesis free" methods. The difference is: for "hypothesis-based" methods, the time series of the stimulus is known, so we can build activation maps corresponding to a certain stimulus; while for "hypothesis-free" methods, the time series of the stimulus is unknown, so no unique explicit activation pattern is available.

For each category, we study feature extraction and feature matching algorithms. Thus, our work consists of 2x2=4 parts, as shown in table 15.1.

Features are represented as a number of voxels (referred to as feature voxels) that have the most significant values in activation assessment. We evaluated several novel approaches to estimate the activation levels, of which the linear FIR model with a Maximum a Posteriori (MAP) smoothing factor clearly beat the main stream GLM with canonical HRF in terms of retrieval performance, in a heterogeneous database. Although SPNN effectively controls the drawbacks of the MAP model with both synthetic data and real data in FIR estimation, it does not improve retrieval effectiveness. One possible reason is that SPNN

Table 15.1: Summary of methods. The boldface is the best in its category.

Parts	**Feature Extraction**	**Feature Matching**
Hypothesis-based (GLM)	Canonical HRF **MAP FIR** SPNN(MAP) FIR LLFOM	Simple Overlap **Fuzzy Overlap** TFIDF LSI PCA Euclidean Manhattan Mahalanobis
Hypothesis-free (ICA)	**LFC** HFC RDC	**MWB Matching** Single Link

does not model undershoot. We argue that the LLFOM model with a single initial point generates comparable result to the canonical HRF, and should be studied in the future with more efficient algorithms and more initial points. The multiple regression does not give significantly better retrieval performance than single regression.

We propose a dual analogy between fMRI retrieval and textual retrieval. Namely, the feature voxels correspond to "terms", and the images correspond to "documents". This analogy brings a useful research perspective to fMRI retrieval, in terms of both effectiveness and efficiency. We mainly evaluate popular metrics such as TFIDF, LSI, Mahalonobis, etc. The "Fuzzy overlap" method we developed gives the best and most robust performance among the tested methods. Note that most information retrieval algorithms can be applied in this framework, and more experiments should be done in the future.

For stimulus-free methods, we propose maximum weighted bipartite matching of selected ICA components with low temporal frequency, which outperforms naive competitors.

In summary, we have conducted a comprehensive investigation, on the major aspects of a content-based fMRI retrieval framework. We have identified several methods which improve the retrieval performance. This study shows that effective

and efficient content-based retrieval of fMRI images is possible. Considering this is a very new research, we are confident that that the performance can be further improved by further work.

Appendix A

A partial review of feature extraction in computer vision

Features are extracted from an image, to preserve necessary information for further processing. By defining features, we can not only improve the processing speed, but also gain some resistance to noise by selecting only major features.

In this section we review several geometric or structural features which may be useful in representing fMRI brain image datasets.

A.1 Edge Detection

Edge means an "abrupt" intensity change along a object surface, in 2D images. Intensity change is our only source of information, and edge is one of the most basic forms of this information.

A.1.1 Marr-Hildreth Algorithm

The classical Marr-Hildreth edge detection method was described in [Marr and Hildreth, 1980]. It is also referred to as "Laplacian edge detection" because it is based on the Laplacian operator applied to intensity values.

Derivatives of Intensity

Let's consider a simple 1D picture in Figure A.1. We show 3 curves plotting the intensity, and first and second derivatives. Intuitively, we have only one edge in

this image, but this edge spreads over an interval. Thus, we select the maximum of the first derivative as the edge point, which means that the second derivative $\partial^2 I/\partial x^2$ is 0.

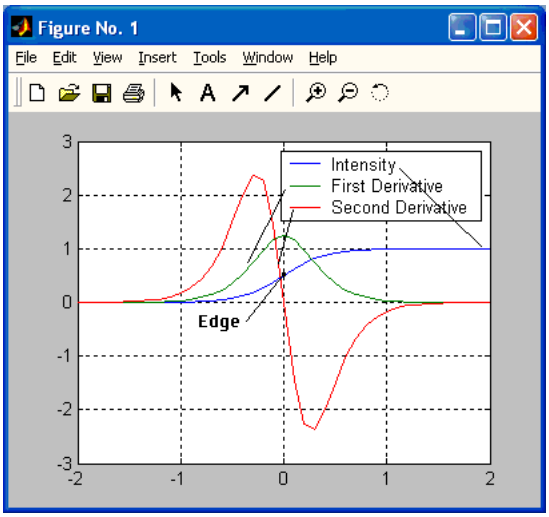


Figure A.1: The edge point is indicated by maximum of first derivative and zero point of second derivative.

Now we come to 2D case. The laplacian

$$\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

is usually considered as an analogue to the second derivative. Thus the edge detection in 2D image is the Laplacian equation

$$\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} = 0 \tag{A.1}$$

In computer implementations, the Laplacian is applied by convolving the image with a square "Laplacian Filter", which is show in Figure A.2.

There is one more problem to solve. A real picture is not so perfect for edge detection. Laplacian edge detection usually uses a Gaussian filter to reduce noises. Since Gaussian filtering is also convolving an array with the image, and according to the convolution rule:

$$L * (G * I) = (L * G) * I \tag{A.2}$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

−1	2	−1
2	−4	2
−1	2	−1

Figure A.2: Three commonly used discrete approximations to the Laplacian filter

the Gaussian and Laplacian are convolved into one single filter called *LoG*. Once *LoG* is formed, both the Gaussian and Laplacian can be applied with only one convolution. *LoG* is also well known as "Mexican hat" for its shape, which is shown in Figure A.3.

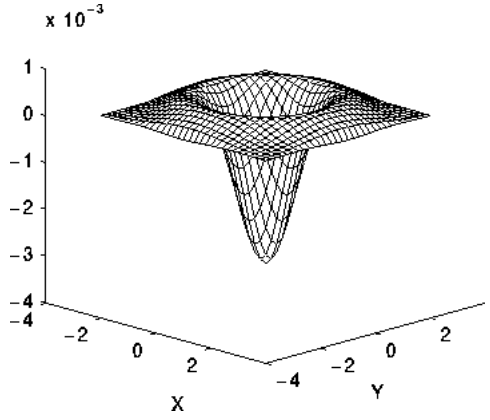


Figure A.3: From [Fisher et al.]. The shape of LoG (Laplacian of Gaussian)

A.1.2 Canny's criteria and Canny's edge detector

Canny's method [Canny, 1986] is related to a so called "optimal filter", which tries to maximize or minimize predefined criteria. Canny defined three criteria for an optimal edge detection filter:

- The detection criterion: no important edges be missed, and no spurious responses should be generated.

- The localization criterion: the distance between the actual and located position of the edge should be minimal.
- The one-response criterion: the number of responses to a single edge should be minimized.

Besides these optimization criteria, Canny designed a optimal edge detection method for a "step edge" - which is defined as a ideal step function whose domain is $(-\infty, +\infty)$ - under white gaussian noise (See Figure A.4).

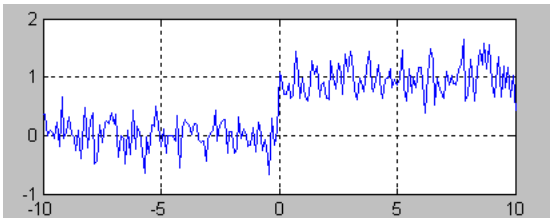


Figure A.4: Canny's step edge model

With quite a bit of deduction, Canny derived a optimal function which is so complicated that we won't list here. It is found that the shape of this optimal filter can be approximated by the first derivative of Gaussian function, which can be efficiently calculated with techniques such as recursive filtering. In 2D pictures, Canny's method is very intuitive since it actually finds the points with maximum gradient magnitude in the direction of the gradient.

Here we give the steps of edge detection:

Non-maximal suppression

We use a Gaussian filter to smooth the image, then do differentiation. According to equation A.2, these two steps can be actually done with one convolution of the derivative of gaussian. To find the local maximum of in the direction of the gradient, we use the directional derivative of Gaussian as a filter.

The gradient of the image can be calculated with

$$\mathbf{n} = \frac{\nabla(G * I)}{|\nabla(G * I)|} \tag{A.3}$$

Then the directional derivative of is

$$G_n = \frac{\partial G}{\partial n} = \mathbf{n} \cdot \nabla G \tag{A.4}$$

The edge location is determined by the zero-crossing of the derivative of G_n in direction of $\mathbf{n}$

$$\frac{\partial}{\partial \mathbf{n}} G_n * I = 0$$

And we finally plug in equation A.4, we have

$$\frac{\partial^2}{\partial \mathbf{n}^2} G * I = 0 \tag{A.5}$$

Equation A.5 is referred to as "non-maximal suppression".

Figure A.5 shows a first derivative of gaussian filter. It's aligned to be parallel to one axis, but it should be rotated to the pixels gradient direction when applying.

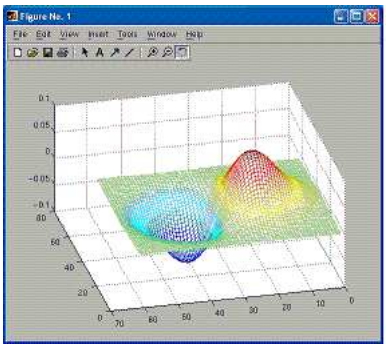


Figure A.5: A picture of the first derivative of gaussian filter

Thresholding

This step corresponds to the 3rd criteria. False edges may be marked as local maximum due to noise. To remove the spurious edges, we threshold the magnitude

of the output of Canny's filter. The magnitude is

$$|G_n * I| \tag{A.6}$$

A.1.3 Active Contour Models(SNAKE)

[Kass et al., 1987] proposed a new way for curve fitting, called "Active Contour Models" or "SNAKE". In addition to setting parameters to adjust a curve to fit the contour of an object, based on image features, SNAKE also considers an "internal energy" of the curve based on the length and curvature. After the user specifies a rough initial contour, the points on the contour moves automatically to places with high intensity contrast, which is likely to be edge points, while at the same time makes the contour as short and smooth as possible. See Figure A.6 for an example.

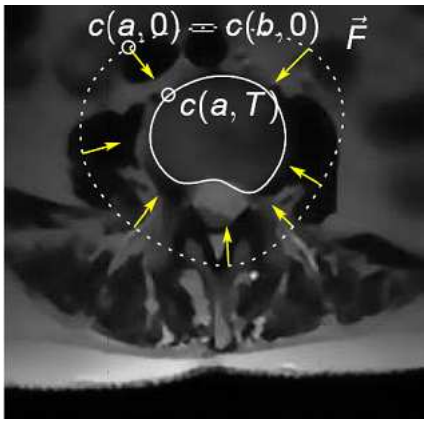


Figure A.6: From [Droske and Azose, 2005]. An example of SNAKE contour evolution.

Energy Minimization

The contour is described with a parametric equation $v(s) = (x(s), y(s))'$, $s \in [0, 1]$ is the parameter of the curve. An "energy" is defined for a contour so that by

minimizing this energy, $v(s)$ will approximate the real boundary of the object.

$$E(v) = S(v) + P(v) \tag{A.7}$$

$S(v)$ is the internal energy for the curve, which is defined as:

$$S(v) = \frac{1}{2}\int_0^1 w_1(s)\left|\frac{\partial v}{\partial s}\right|^2 + w_2(s)\left|\frac{\partial^2 v}{\partial s^2}\right|^2 ds \tag{A.8}$$

The first term is called "tension" or "elasticity", which limits the length of a contour. The second term is "rigidity" or "stiffness", it ensures smoothness of a contour.

The second term in equation A.7 is called "external energy", which comes from the image.

$$P(v) = \int_0^1 P(v(s))ds \tag{A.9}$$

For edge detection, P(v(s)) is defined as

$$P(x,y) = -c\,|\nabla[G_\sigma * I(x,y)]| \tag{A.10}$$

The assumption is, the real boundary often is smooth at some scale. A very wiggling curve is very likely due to noise. By adjusting the component weights w_0, w_1, and c, we can balance the likelihood of introducing noise and that of missing a real boundary point.

A.1.4 Level Set Method and Implicit Active Contour

The dynamic snake shows us the potential of a moving boundary according to internal and external forces. External forces dominate the main direction of the boundary movement, while the internal forces suppress the noises.

Since Osher and Sethian published their important paper [Osher and Sethian, 1988], the level set method has been used in many areas. In computer vision, the level set method combined with the active contour model [Caselles et al., 1995] has been recognized as a very promising method for object detection. Compared to

original SNAKE, the level set method has the advantage of automatic topological adaptivity. The disadvantage is that it does not have an explicit parametric model.

Level Set Method

The level set method was originally designed as a method to calculate the propagating interfaces like wave and flame. We define a function $\varphi(x,t)$ on $R^n \times t$, where R^n is an n dimensional space, and t is the time axis. The interface at time t is defined by

$$\varphi(x,t) = 0 \tag{A.11}$$

At the beginning, some function value φ is considered as a continuous landscape. The points on a closed loop are selected as the contour with $\varphi(x,t) = 0$, the points inside this loop are less than 0, while the points outside the loop are greater than 0. Level sets can be built for φ values.

The motion of the interface is defined by a PDE:

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \varphi = 0 \tag{A.12}$$

$\frac{\partial \varphi}{\partial t}$ on the left is the time rate of change, $\nabla \varphi$ is the gradient of the function value over the space, $\mathbf{v}$ is the "speed" of the movement of the "wave". So, if we initialize the function value φ of every grid point in the space, such that the initial boundary points has $\varphi = 0$, while the rest of the space has positive and negative value according to the level set position to the boundary, then the boundary will move over the time according to equation A.12. Figure A.7 shows the process of the front evolution:

In [Osher and Sethian, 1988], a numerical discrete scheme is presented for front propagation, to handle "shock" and "fanout".

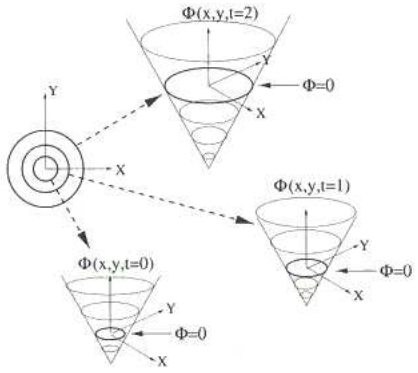


Figure A.7: From [Sethian, 1999]. The level set method. The boundary is marked by the 0 values.

Geometric Active Contour

The geometric active contour method in [Caselles et al., 1995] can be summarized by equation A.13. To be consistent with the [Caselles et al., 1995], we replace the function name φ with u.

$$\frac{\partial u}{\partial t} = g(x)\,|\nabla u|\,(div(\frac{\nabla u}{|\nabla u|}) + v) \quad (t,x) \in [0,\infty) \times R^2 \tag{A.13}$$

$$u(0,x) = u_0(x) \quad x \in R^2 \tag{A.14}$$

where

$$g(x) = \frac{1}{1 + (\nabla G_\sigma * g_0)^2} \tag{A.15}$$

This is a PDE with initial conditions. Comparing it to equation A.12, we see the speed factor **v** is replaced by

$$g(x)[div(\frac{\nabla u}{|\nabla u|}) + v]$$

. $g(x)$ is the factor corresponding to "external energy" in classical SNAKEs, if there is no intensity change, this factor is 1, and it approximates 0 when the gradient become larger. Thus, the front will tend to stop at the pixels with large gradient, which is likely to be an edge. $div(\frac{\nabla u}{|\nabla u|})$ is the curvature, Larger curvature will cause larger speed. v is a constant to guarantee the motion of the boundary.

In each iteration, the value at each voxel/pixel is updated based on equation A.13, and the positions of the voxel/pixels with 0 values change, which corresponds to the moving of boundary.

A.1.5 Complexity of edge detection algorithms

If the number of voxels/pixels is n, then the Laplacian method and Canny's method takes time $O(n)$ since they are both based on simple filtering and thresholding. For the parametric active contour method, the time depends on the length of the contour and the number of iterations, since it computes internal and external energy for the contour. For the level set active contour method, the time complexity is much higher, because in each iteration the φ value of the whole volume needs to be updated. To solve this problem, narrow band method and fast marching method have been proposed. They are not discussed here.

A.2 Ridge detection

In fMRI study, the resolution is quite low and edge detection will suffer from the fuzziness of the image. We consider there another type of features called ridge. As the word suggests, a ridge is like a ridge in a mountain. It is a bunch of points aligned in a local line-like structure and is the local maximum in the direction perpendicular to the line. In [Koenderink, 1990], a more meaningful description of ridge is defined, which enables the detection of the ridges: specifically, the curvature of contours reaches a local maximum on ridge points. See Figure A.8 for demonstration of ridges in 2D map.

In a 3D case, the situation is similar, but a little more complicated. In 3D, the contours become isosurfaces. We describe the curvature of any point with two "principal curvatures" of the isosurface. For every plane which passes the normal of a point, there will be a intersection curve with the surface. The curvature of

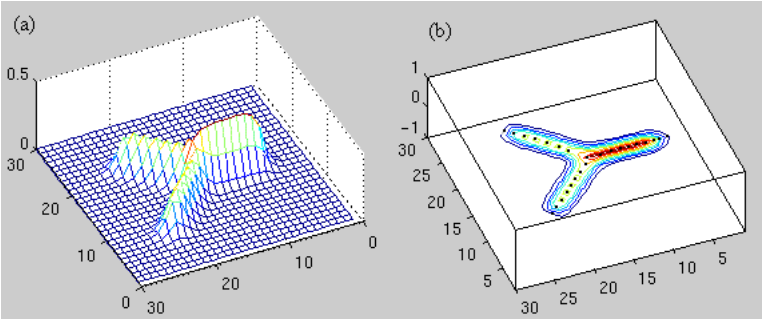


Figure A.8: A simulated landscape and its ridges. (a) shows the a landscape with 3 ridges. (b) shows that the ridges points have local maximal curvatures (the black dots)

this curve at the point is called normal curvature. see Figure A.9.

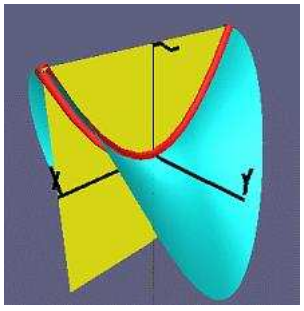


Figure A.9: An example of normal curvature

The maximum and the minimum normal curvatures are called principal curvatures. The planes for maximum principal and minimum principal are perpendicular, A ridge point is a point at which either of two principle curvatures reaches a maximum.

A.2.1 Monga's method

[Monga and Benayoun, 1995] described a analytical method to locate ridge points in 2D and 3D. Consider the 2D case first. For each pixel/voxel, we represent the gradient as $\vec{g} = (I_x, I_y)$, where I_x and I_y are derivatives of intensity, and the

tangent of the contour as $\vec{t}$. We have

$$\vec{g} \cdot \vec{t} = 0$$

Let s be the arc length parameter, we have

$$t = (\frac{\partial x}{\partial s}, \frac{\partial y}{\partial s})$$

and

$$\frac{d(\vec{g} \cdot \vec{t})}{ds} = \frac{d\vec{g}}{ds} \cdot \vec{t} + \vec{g} \cdot \frac{d\vec{t}}{ds} = 0 \tag{A.16}$$

The curvature is:

$$\frac{d\vec{t}}{ds} = k\vec{n} = k\frac{\vec{g}}{\|\vec{g}\|} \tag{A.17}$$

The differentiation of the gradient along the curve is:

$$\frac{d\vec{g}}{ds} = \frac{\partial \vec{g}}{\partial x}\frac{dx}{ds} + \frac{\partial \vec{g}}{\partial y}\frac{dy}{ds} = H\vec{t} \tag{A.18}$$

where H is hessian

$$H = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix}$$

By combining equation A.16, A.17 and A.18, we have

$$k = \frac{-t^T H t}{\|\vec{g}\|} \tag{A.19}$$

The 3D case is a little more complicated. First of all, for each point, there is a tangent surface, not a tangent line. If we choose a particular direction $\vec{t}$, the curvature is in the similar form

$$k_{\vec{t}} = -\frac{t^T H t}{\|\vec{g}\|} \tag{A.20}$$

We want to find a direction in which the curvature reaches its extrema. Let $\vec{a}$ and $\vec{b}$ be an orthogonal basis of the tangent plane, and θ be the angle from $\vec{a}$ to

$\vec{t}$, so $\vec{t} = \vec{a}\cos\theta + \vec{b}sin\theta$. When curvature reaches the extremum, we have

$$\frac{dk_{\vec{t}}}{d\theta} = 0 \Leftrightarrow \quad \text{(A.21)}$$

$$\frac{\partial t^T H t}{\partial \theta} = 0 \Leftrightarrow \quad \text{(A.22)}$$

$$\frac{\partial}{\partial \theta}(\vec{a}^T H \vec{a}\cos^2(\theta) + \vec{b}^T H \vec{b}\sin^2(\theta) + 2\vec{a}^T H \vec{b}\sin(\theta)\cos(\theta) = 0 \Leftrightarrow \quad \text{(A.23)}$$

$$(\vec{a}^T H \vec{a} - \vec{b}^T H \vec{b})\sin 2\theta + 2\vec{a}^T H \vec{b}\cos 2\theta = 0 \Leftrightarrow \quad \text{(A.24)}$$

$$\tan 2\theta = \frac{2\vec{a}^T H \vec{b}}{\vec{a}^T H \vec{a} - \vec{b}^T H \vec{b}} \quad \text{(A.25)}$$

Thus theta can be computed using the arctan function, and the principal directions are decided for equation A.20.

To find the ridge location in the 2D case, term A.19 should have a zero derivative with respect to s.

$$\frac{3(t^T H t)(g^T H t) - \|\vec{g}\|^2 t^T \begin{pmatrix} t^T H_x t \\ t^T H_y t \end{pmatrix}}{\|\vec{g}\|^3} = 0 \quad \text{(A.26)}$$

And for 3D case, term A.19 should have zero derivative along the principal curvature direction:

$$\frac{\|\vec{g}\|^2 t_1^T \cdot \begin{pmatrix} t_1^T H_x t_1 \\ t_1^T H_y t_1 \\ t_1^T H_z t_1 \end{pmatrix} - (t_1^T H t_1)(t_1^T H g)}{\|\vec{g}\|^3} = 0 \quad \text{(A.27)}$$

The derivatives like H and H_x can be computed with convolution of corresponding derivatives of smoothing filters like Gaussian or Deriche filter [Deriche, 1987].

A.3 Skeletonization

To skeletonize is to find the "back bone" in a shape. In Figure A.10, we can see a sample object and its "Skeleton". The idea of skeletonization came from Blum's "Medial Axis Transformation" (MAT) [Blum, 1967].

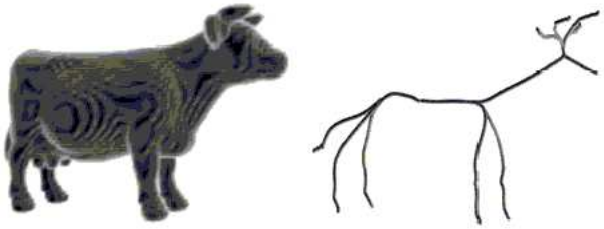


Figure A.10: From [Cornea et al.]. An example skeleton.

Skeleton finding is the same as ridge finding in the sense of finding the centerline. The difference is, skeletonization usually find the centerline in an object described by its boundary points, while ridge finding seeks the centerline in an volume. However skeletonization can be done by finding ridges in the distance map we will discuss below. The brain image does not have explicit boundaries, but the boundaries can be built with edge detection methods, so it is still potentially valuable.

Skeletonization has received great attention because of its abstraction ability. Such abstraction/simplification makes efficient matching possible. Application of skeletonization can be found in Siddiqi et al. [1998], Sundar et al. [2003].

Skeletonization methods can be categorized into 4 categories:

- Simulation of grass fire.
- Analytical computation of medial axis.
- Topological thinning.
- Medial axis extraction from distance map.

A.3.1 Grass fire method

The grass fire method takes its name from the idea that when the fire burns from the boundary of a grassland, the skeleton will be the points where flames

finally meet. This is also known as "shock detection" since it's similar to the shock in study of the wave equation. To some degree this reminds us of the front evolution methods like the active contour method and level set method. In fact some of the work on grass fire came from combination of those methods [Xia, 1989, Leymarie and Levine, 1992, Telea and van Wijk, 2002]. Here we introduce the work of [Telea and van Wijk, 2002, Telea et al., 2004] based on the "fast marching method" [Sethian, 1996].

Fast Marching Method (FMM)

The Fast Marching Method is a method of tracking a propagating front. The idea is a little bit like Dijkstra's algorithm. Suppose an initial boundary, consisting of a bunch of grid vertices, is defined. We define the "arriving time" $T(x, y)$ of the front at these vertices to be 0. Suppose the speed of the front evolution $F(x)$ is always positive, then we can mark the "arriving time" of surrounding vertices based on the speed $F(x)$. Of course a vertex can be reached from different directions, we mark it with the earliest arriving time.

Skeletonization based on FMM

At the beginning, the boundary of the object (which we are going to skeletonize) is initialized. Every vertex on this boundary is marked with a increasing label. When the boundary is "marching" to the center, the vertices are labeled with the unique label of its source. Some of the vertices may be reached from different sources. If the difference between source labels is greater than 2, then the vertex is a skeleton point because it is actually the meeting point of grass fire from non-neighboring sources. Figure A.11 shows the idea.

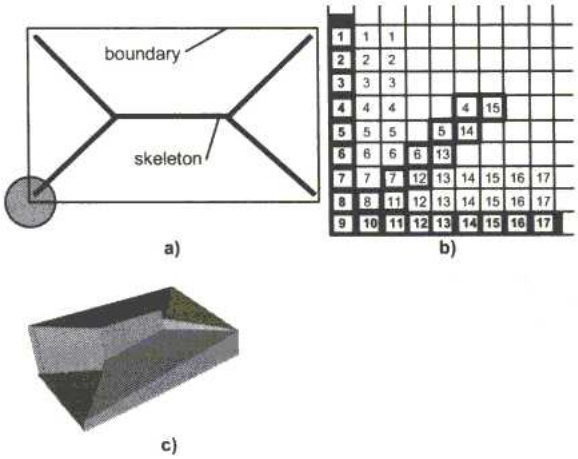


Figure A.11: (From [Telea and van Wijk, 2002].) Skeletonization based on fast marching algorithm. When the marching of two non-neighboring source vertex meet, the meeting point is marked as a skeleton point.

A.3.2 Voronoi skeleton

Voronoi Diagram

The Voronoi diagram [de Berg et al., 2000] is a important partition scheme in computational geometry. In this scheme, a number of vertices are defined in a shape, and the shape is partitioned so that every partition contains a vertex and the points that are closest to this vertex. Figure A.12 is an example.

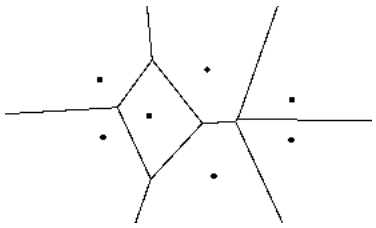


Figure A.12: Voronoi Diagram

Discrete Voronoi Skeleton

The Voronoi skeleton can be defined as the boundaries of Voronoi partitions. In other words, it's the intersection between two voronoi partitions. If we consider every point on the boundary as a vertex, then the Voronoi diagram can be built in $O(n \log n)$ time [de Berg et al., 2000]. However, as we show in Figure A.13, these voronoi edges will cover the whole image. Most of them are not "salient". It is also subject to noise on the boundary.

The discrete Voronoi skeleton [Ogniewicz and Kuebler, 1995] defines the "importance" of a Voronoi point as the "Potential Residual", which we will describe below. First, we look at the concept of anchor points. Each point on a Voronoi edge has usually two closest points on the boundary, these points are called "anchor points". The potential residual is the length of the minimum length from one anchor point to another along the boundary. See Figure A.13, p_A and p_A^* are anchor points for voronoi point m_A. The potential residual $R(m_a)$ is length of the dashed line which connects p_A and p_A^*. The similar cases for m_B and m_C.

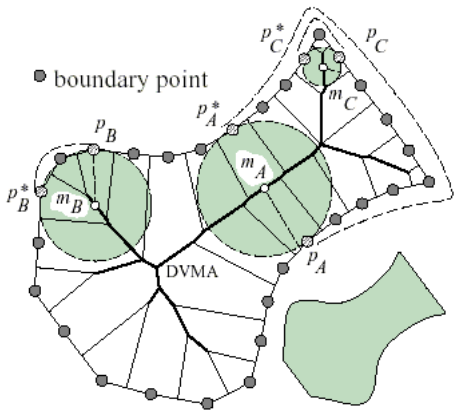


Figure A.13: From [Ogniewicz and Kuebler, 1995]. Potential residuals.

To eliminate the trivial skeletons, we set a threshold on the potential residual R. In other words, we only consider a point with high enough potential residual

as a part of skeleton.

A.3.3 Skeleton based on potential field

The above two methods were presented as measures for dealing with 2D images. And this work has not been not extended to 3D objects, to the best of my knowledge. However, finding the "center line" in 3D object has attracted a lot of attraction in recent years. [Chuang et al., 2000] proposed a method based on "force field". [Cornea et al.] extended the idea. Every vertex at a boundary of the object is considered as a particle with mass, and can exert a "Quasi-gravity force" defined as:

$$\vec{F} = \frac{\vec{CP}}{|CP|^m} \tag{A.28}$$

where C is the vertex exerting the force, and P is the observation position in the force field. We see that the force is repelling instead of attracting. This is to allow the "flow" that we describe later.

We see that this formula generalizes the gravitational force by making the power of distance variable. Such an ensemble of vertices generates a vector field of pseudo-anti-gravity. Very interestingly, if a particle is put into this vector field and follow the pesudo-anti-gravity force vector (assuming no inertia), the trajectory is pretty much a centerline of the boundary. This is because, in the plane perpendicular to the force vector, the total force is 0. Since the force is determined by the distance from the force exerting vertices, this means that the distances to vertices have roughly equal projections on this plane, which makes the point a center line in this plane.

There are several problems with [Chuang et al., 2000]. First, the starting point of the trajectory is not clearly defined. In [Chuang et al., 2000], orbits just start from "convex corners" of the object boundary, and follow the force vector until it stops. Second, the connection between different parts of skeleton is not

efficiently calculated. They connect each pair of critical points with straight lines, and let the line evolve within an active contour model. Some of the connection are discarded later if they don't look good.

[Cornea et al.] extended the idea of force field, and proposed a simpler and more robust method to deal with it. First, they build a force vector field for each grid point inside the object volume. Then, two types of "critical points" are defined: attracting points and saddle points. An attracting point is the stationary balance point that the total force at it is 0, and the force will oppose any small shift. A saddle point is a point where the total force is 0, but is stable only in some directions. The type of critical point can be determined by the jacobian matrix. If all the eigenvalues of the jacobian matrix are negative, it is an attracting point. Otherwise it's a saddle point, and the stationary and non-stationary directions can be read from the eigenvectors.

Once we have all the critical points available, we can start from saddle points, and follow outgoing directions. This flow will bring us to attracting points. The paths of the flow become the parts of the skeleton.

A.3.4 The Shock Graph

A skeleton is a reduction from 2D or 3D data to 1 local degree of freedom. Although the data has been abstracted a lot, some research seeks to do more. [Siddiqi et al., 1998] extended the work in [Blum, 1984], and developed a way to make a directed acyclic graph (DAG) from the skeleton to make the abstraction more robust to the noise in the shape.

Shock [Kimia et al., 1995] is a term introduced from mathematics when the evolution equation

$$\frac{\partial \vec{C}(x,y)}{\partial t} = (1 + \alpha k)\vec{N}$$

was studied. $\vec{C}(x,y)$ is a point on the propagating front, k is the curvature, and N

is the normal direction. Roughly speaking, shock can be understood when wave occurring from non-neighboring sources collide. Here we assume α is 0, so the evolution of the wave front at this point is simply following the normal direction, until they meet with the waves from other directions. This is the same concept of the grass fire skeleton process. We can think the shock points as the skeleton points.

[Kimia et al., 1995] divides the shocks into 4 types shown in Figure A.14.

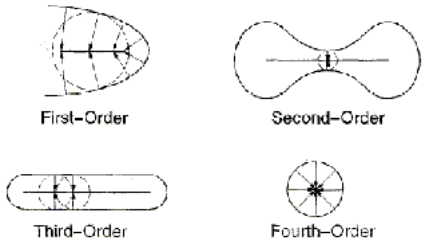


Figure A.14: From [Siddiqi et al., 1998]. Shock types. Type 1 is called first-order shock or protrusion, which is shown in [Kimia et al., 1995] as the result of local curvature extreme. Type 2 (second-order) is like a "neck", it's a thinner part in between two thicker parts. Type 3 (third-order) happens when the neighboring segments have the same thickness. Type 4 (fourth-order) are the points where all the evolving boundary points finally merge.

Figure A.15 shows an example of the shocks in a graph.

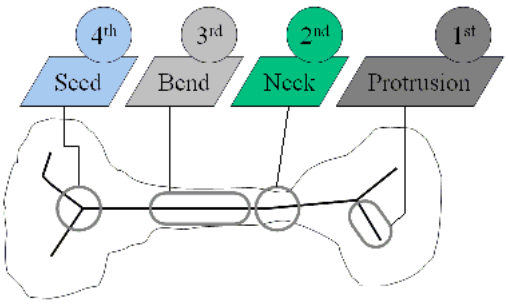


Figure A.15: From [Siddiqi et al.] An example of different sock types in an object.

The shock graph is built according to the time order of "grass fire" processing.

A shock point is generated earlier will be closer to leaves. In A.15 we see the segment of 3rd type shock consists of the same type of shock points. In shock graph, such neighboring same type of shock points are represented with one single node. The shock graph of Figure A.15 is shown in Figure A.16:

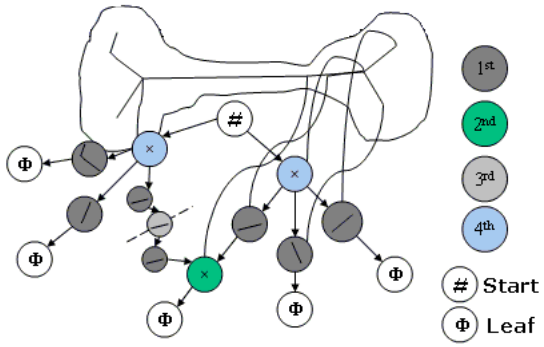


Figure A.16: From [Siddiqi et al.]. The shock graph of Figure A.15

A.4 Contour Tree

The contour tree is an abstraction of topological relations among contours. The input of contour tree algorithms is usually a (k-dimension) mesh with a function value at each mesh point. A typical 2D case is a terrain, where each mesh point has height as its function value. The algorithms output simple tree structure which allows us to trace important spatial structural changes. Figure A.17 shows an example of a contour tree.

If the only available data is the function value at sample points, delauney triangulation meshes can be obtained with randomized increment algorithms as described in [de Berg et al., 2000]) in $O(n \log n)$ time.

From Figure A.17, we can see the following properties of contour trees:

- A local maximum or a local minimum corresponds to a leaf node of the tree. The degrees of these nodes are 1.

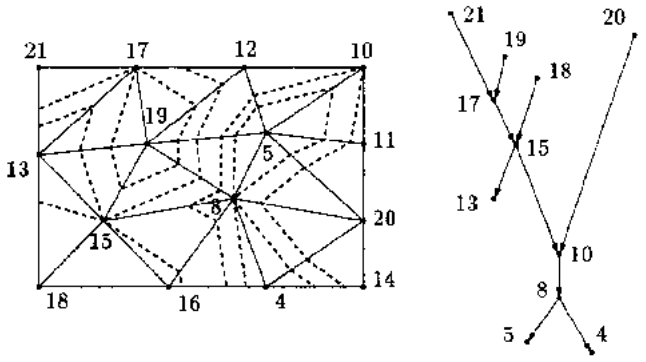


Figure A.17: From [van Kreveld et al., 1997]. An example contour tree.

- A saddle point corresponds to an internal node of the tree, the degree is at least 3. On a saddle point the topology of the contours may change. Two contours may merge, a contour may split, and so on.

To determine if a vertex is a extremum or a saddle point, one needs to look at values of all the neighboring vertices. If all are less than (or greater than) the current vertex, it's a maximum (or minimum) vertex. If the neighboring value are greater or less than current value alternately, then it's a saddle point.

[van Kreveld et al., 1997] proposed following plane sweeping method which runs in generally $O(n^2)$ time where n is the number of mesh cells. One can prove that n is roughly proportional to the number of vertices (de Berg et al. 2000 [de Berg et al., 2000]).

First, all the vertices are sorted by the function value. Then the plane is swept from the highest value to the lowest value. The algorithms maintain current contour components by using "active" cells as we describe later. The sweeping stops at each vertex, and handles the following possible situations:

- If it's a maximum vertex, a new contour component is generated. A new tree node is generated. And incident cells of the vertex become active.

- If it's a minimum vertex, the corresponding contour component is eliminated. A new tree node is generated. Its incident cells of the vertex become inactive.

- If it's a saddle vertex, a new node is generated, contours are splitted or merged according to the situation.

The sorting takes $O(n \log n)$ time. In the process of sweeping, we need to look at the neighboring cells of the current vertex. In worst case, this takes $O(n)$ time for each vertex. So the total time is $O(n^2 + n \log n) = O(n^2)$.

[van Kreveld et al., 1997] also shows that in the 2D case, the algorithm can run in $O(n \log n)$ time. [Tarasov and Vyalyi, 1998] showed an algorithm running in $O(n \log n)$ time for 3D data. [Carr et al., 2003] extended Tarasov and Vyalyi [1998], and proposed an algorithm that runs in $O(m \log m + n\alpha(n))$, where α is an extremely slowly growing inverse of Ackermann's function, and m is the number of vertices.

Appendix B
Performance Evaluation

After a retrieval system is implemented, we will need to evaluate the performance of our method. For any query, we have a ranked list based on the similarity computed by the algorithm. We also have some ranked list based on prior knowledge of each dataset, such as whether the datasets are from same subject or under the same conditions. Such common properties should make more similar datasets. The more consistent these two ranked lists are, the better performance of the algorithms.

To evaluation the consistency of ranked lists, we can use Spearman's rank correlation [Lehmann and D'Abrera, 1998] and Kendall's tau rank correlation [Kendall, 1970].

B.1 Spearman's Rank Correlation

Spearman's rank correlation is defined as:

$$r_s = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{B.1}$$

where d_i is the rank difference of element i in two lists.

It can be shown that the distribution of r_s reaches a maximum value when the two ranked lists are exactly the same, and reaches a minimum value when they are in opposite order. The average of r_s over all permutations is 0.

B.2 Kendall's Tau Rank Correlation

Suppose we have two ranked lists S and T for the same set of elements. The ranks of the ith element can be recorded in the pair $(S_i, T_i), i = 1, \cdots, n$. If for any $S_i < S_j$ and $(T_i < T_j)$, we say the pairs (S_i, T_i) and (S_j, T_j) are *concordant*, which means element i and j have the same order in two ranked list. On the other hand, if for $S_i < S_j$ there is $(T_i > T_j)$, then i and j are called discordant.

For a set with n elements, the number of all the possible element pairs (i, j) is $\frac{n(n-1)}{2}$. If we denote the number of concordant pairs by n_c, and denote the number of discordant pairs by n_d, the Kendall's tau is defined as:

$$\tau = \frac{n_c - n_d}{n(n-1)/2} \tag{B.2}$$

Note that $\frac{n_c}{n(n-1)/2}$ is the probability of concordant element pairs, and $\frac{n_d}{n(n-1)/2}$ is the probability of concordant element pairs. Thus Kendall's tau describes the degree of concordance between two ranked lists. If the two ranked lists are identical, Kendall's tau gives value 1. If the two ranked lists are not correlated, Kendall's tau gives value 0. If they are in complete revered order, Kendall's tau is -1.

Appendix C
MAP estimation of FIR weights

C.1 MLE parameter estimation with Gaussian noise

Let $Y = X\beta + \epsilon, \epsilon\tilde{N}(0,\Sigma)$, where $Y = [y_1, y_2, ..., y_N], \epsilon$ The likelihood of the obervation Y given β is:

$$\begin{aligned} P(Y|\hat{\beta}) &= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y_i - x_i^T\hat{\beta})^2}{2\sigma^2}} \\ &= (\frac{1}{\sqrt{2\pi\sigma^2}})^N e^{-\frac{\sum_{i=1}^{N}(y_i - x_i^T\hat{\beta})^2}{2\sigma^2}} \\ &= (\frac{1}{\sqrt{2\pi\sigma^2}})^N e^{-\frac{(Y - X\hat{\beta})^T(Y - X\hat{\beta})}{2\sigma^2}} \end{aligned}$$

Thus, Miximizing $P(Y|\hat{\beta})$ is equivalent to minimizing $D = (Y - X\hat{\beta})^T(Y - X\hat{\beta})$, and this is the objective function of least square approximation, which can be solved by linear regression. Specifically, when D reaches extremum,

$$\begin{aligned} \frac{\partial D}{\partial \hat{\beta}} &= \frac{\partial}{\partial \hat{\beta}}(Y^TY - 2Y^TX\hat{\beta} + \hat{\beta}^TX^TX\hat{\beta}) \\ &= -2X^TY + 2X^TX\hat{\beta} = 0, \end{aligned} \tag{C.1}$$

and that gives the MLE estimate $\hat{\beta}_{MLE} = (X^TX)^{-1}X^TY$.

C.2 MAP estimation with Gaussian noise and Gaussian Prior

Suppose the β has joint Gaussian prior distribution $P(\hat{\beta})\tilde{N}(0,\Sigma)$. According to Bayesian rule,

$$P(\hat{\beta}|Y) = \frac{P(Y|\hat{\beta})P(\hat{\beta})}{P(Y)}.$$

There is nothing we can do to the denominator $P(Y)$, so Maximizing the posterior probability $P(\hat{\beta}Y)$ is equivalent to maximize the numerator:

$$\begin{aligned} P(Y|\hat{\beta})P(\hat{\beta}) &= (\frac{1}{\sqrt{2\pi\sigma^2}})^N e^{(Y-X\hat{\beta})^T(Y-X\hat{\beta})} \cdot \frac{1}{\sqrt{(2\pi)^k det\Sigma}} e^{-\frac{1}{2}\hat{\beta}\Sigma^{-1}\hat{\beta}} \\ &= \frac{1}{\sqrt{2\pi\sigma^2}^N \sqrt{(2\pi)^k det\Sigma}} e^{(Y-X\hat{\beta})^T(Y-X\hat{\beta})-\frac{1}{2}\hat{\beta}\Sigma^{-1}\hat{\beta}}, \end{aligned}$$

which is equivalent to minimizing the exponential:

$$-Y^TY + 2Y^TX\hat{\beta} - \hat{\beta}(X^TX + \frac{1}{2}\Sigma^{-1})\hat{\beta}.$$

Similar to Eq C.1, we have

$$\hat{\beta}_{MAP} = (X^TX + \frac{1}{2}\Sigma^{-1})^{-1}X^TY.$$

Bibliography

A. Aron, H. Fisher, D.J. Mashek, G. Strong, H. Li, and L.L. Brown. Reward, motivation, and emotion systems associated with early-stage intense romantic love. *J Neurophysiol*, 94:327–337, 2005.

B. Bai and P. Kantor. A shape-based finite impulse response model for functional brain images. In *Proceedings of the 4th IEEE International Symposium on Biomedical Imaging*, 2007.

B. Bai, P. Kantor, N. Cornea, and D. Silver. Ir principles for content-based indexing and retrieval of functional brain images. In *Proceedings of the 15th ACM Conference on Information and Knowledge Management (CIKM06)*, 2006.

B. Bai, P. Kantor, N. Cornea, and D. Silver. Toward content-based indexing and retrieval of functional brain images. In *Proceedings of the (RIAO07)*, 2007a.

B. Bai, P. Kantor, and A. Shokoufandeh. The effectiveness of finite impulse response model in content-based fmri image retrieval. In *submission*, 2007b.

B. Bai, P. Kantor, A. Shokoufandeh, and D. Silver. fmri brain image retrieval based on ica components. In *submission*, 2007c.

C. Beckmann and S. Smith. Probabilistic independent component analysis for functional magnetic resonance imaging. *IEEE Transactions on Medical Imaging*, 23(2), 2004.

A.J. Bell and Terrence J. Sejnowski. An information-maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1004–1034, 1995.

C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.

F. Bloch, W. W. Hansen, and M. Packard. Nuclear induction. *Physical Review*, 69:127, 1946.

H. Blum. *A Transformation for Extracting New Descriptors of Shape*. MIT Press, Cambridge, MA, 1967.

H. Blum. Biological shape and visual science. *J. Theor. Biol.*, 38:205–287, 1984.

J.A. Bondy and U.S.R. Murty. *Graph Theory with applications*. Macmillan, London, 1976.

E. Bullmore, J. Fadili, M. Breakspear, R. Salvador, J. Suckling, and M. Brammer. Wavelets and statistical analysis of functional magnetic resonance images of the human brain. *Statistical Methods in Medical Research*, 12(5):375 – 399, 10 2003.

R.B. Buxton, E.C. Wong, and L.R. Frank. Dynamics of blood flow and oxygenation changes during brain activations: The balloon model. *Magnetic Resonance in Medicine*, 39:855–864, 1998.

V.D. Calhoun, T. Adali, G.D. Pearlson, and J.J. Pekar. Spatial and temporal independent component analysis of functional mri data containing a pair of task-related waveforms. *Human Brain Mapping*, 13:43–53, 2001.

V.D. Calhoun, T. Adali, L.K. Hansen, J. Larsen, and J.J. Pekar. Ica of functional mri data: An overview. In *4th Int. Sym. on Indep. Comp. Analy. and Blind Signal Sep*, 2003.

J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8, 1986.

H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.

V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *IEEE International Conference in Computer Vision*, pages 694–699, 1995.

J. Chuang, C. Tsai, and M. K. Skeletonization of three-dimentional object using generalized potential field. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1241–1251, 2000.

N.D. Cornea, D. Silver, X. Yuan, and R. Balasubramanian. Computing hierarchical curve-skeletons of 3d objects. *Visual Computer (to appear)*.

R.W. Cox. Afni: software for analysis and visualization of functional magnetic resonance neuroimages. *Comput. Biomed. Res.*, 29, 1996.

R. V. Damadian. Tumor detection by nuclear magnetic resonance. *Science*, 171 (1151), 1971.

M. de Berg, M. van Kreveld, M. Overmans, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications, 2nd rev. ed.* Springer-Verlag, Berlin, 2000.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990. URL `http://citeseer.ist.psu.edu/deerwester90indexing.html`.

R. Deriche. Using canny's criteria to derive a recursively implemented optimal edge detector. *Internat. J. Vision*, 1987.

C. Dickens. *A Tale of Two Cities*. Chapman and hall, 1859.

M. Droske and B. Azose. A tutorial on snakes and active contours. In *http://www.math.ucla.edu/~droske/snakes.pdf*, 2005.

J. Edmonds. Paths, trees and flowers. *Canad. J. Math*, 17:449–467.

R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Image processing learning resources. In *http://homepages.inf.ed.ac.uk/rbf/HIPR2/.*

R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

J. Ford, H. Farid, F. Makedon, L.A. Flashman, T.W. McAllister, V. Megalooikonomou, and A.J. Saykin. Patient classification of fmri activation maps. In *6th Annual International Conference on Medical Image Computing and Computer Assisted Intervention*, 2003.

F. Forzaneh, S.J. Riederer, and N.J. Pelc. Analysis of t2 limitations and off-resonance effects on spatial resolution and artifacts in echo-planar imaging. *Magn. Reson. Med.*, 14:123–139, 1990.

J. Fox. *Linear statistical models and related methods.* John Wiley & Sons, Inc., 1984.

R.S.J. Frackowiak, K.J. Friston, C.D. Frith, R.J. Dolan, C.J. Price, S. Zeki, J. Ashburner, and W. Penny. *Human Brain Function (2nd Edition).* Elsevier Academic Press, 2004.

K.J. Friston, P. Jezzard, and R. Turner. Analysis of functional MRI time-series. *Human Brain Mapping*, 1:153–171, 1994. URL `/spm/doc/papers/fMRI_4/`.

K.J. Friston, A. Mechelli, R. Turner, and C.J. Price. Nonlinear responses in fmri: the balloon model, volterra kernels, and other hemodynamics. *Neuroimage*, 12(4):466–477, 2000. URL `http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve\&db=pubmed\&dopt=Abstract\&list_uids=10988040`.

K.R. Gabriel. Generalised bilinear regression. *Biometrika*, 85(3):689–700, 1998.

K.R. Gabriel and S. Zamir. Lower rank approximation of matrices by least squares with any choice of weights. *Technometrics*, 21(4):489–498, 1979.

C. Goutte, F.A. Nielsen, and L.K. Hansen. Modeling the haemodynamic response in fmri using smooth fir filters. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, 19(12), 2000.

J.D. Greene, R.B. Sommerville, L.E. Nystrom, J.M. Darley, and J.D. Cohen. An fmri investigation of emotional engagement in moral judgment. *Science*, 293, 2001.

H.J.A.M Heijmans. *Morphological Image Operators. Advances in Electronics and Electron Physi cs*. Boston: Academic Press, 1994.

J.D. Van Horn, J.S. Grethe, P. Kostelec, J.B. Woodward, J.A. Aslam, D. Rus, Rockmore, and M.S. Gazzaniga. The functional magnetic resonance imaging data center (fmridc): the challenges and rewards of large-scale databasing of neuroimaging studies, 2001.

J.P. Hornak. the basics of fmri. In *http://www.cis.rit.edu/htbooks/mri/inside.htm*, 2005.

D. Hu, L. Yan, Y. Liu, Z. Zhou, K.J. Friston KJ, C. Tan C, and D. Wu. Unified spm-ica for fmri analysis. *Neuroimage*, (3):746–55, 2005.

A. Hyvarinen. *Survey on Independent Compoenent Analysis*, pages 94–128. 1999.

A. Hyvarinen. Fastica matlab package, 2005.

A. Hyvarinen and E. Oja. Independent compoenent analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

A. Hyvarinen, J. Karkunen, and E. Oja. *Independent Compoenent Analysis*. John Wiley and Sons, 2001.

U. Ibraev. *Imposing Graph Structures on Space-Time Densities For Indexing and Retrieval*. PhD thesis, 2005.

W. James. *Principles of Psychology*. 1890.

C. Jutten and J. Herault. Blind separation of sources. 1. an adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24(1):1–10, 1991.

M. Kass, A. Witkin, and D. Terzopoulos. Snakes - active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

M. G. Kendall. *Rank Correlation Methods*. Griffin, London, 1970.

H. Kim and H. Park. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. *Bioinformatics*, 23(12):1495–1502, 2007.

B.B. Kimia, A. Tannenbaum, and S.W. Zucker. Shapes, shocks, and deformations, i. *International Journal of Computer Vision*, 15:189–224, 1995.

J. J. Koenderink. *Solid Shape*. MIT Press, Boston, 1990.

H. Kuhn. The hurigarian method for the assignment problem. *Naval Res. Logist. Quart.*, 2:83–97, 1955.

A. Kumar, D. Welti, and R.R. Ernst. Nmr fourier zeugmatography. *J. Magn. Reson*, 18:69–83, 1975.

K. K. Kwong, J. W. Belliveau, D. A. Chesler, I. E. Goldberg, R. M. Weisskoff, Be. P. Poncelet, D. N. Kennedy, B. E. Hoppel, M. S. Cohen, and R. Turner et al. Dynamic magnetic resonance imaging of human brain activity during primary

sensory stimulation. In *Proc. Nat. Academy of Sciences USA*, volume 89, pages 5675–5679, 1992.

S. LaConte, S. Strother, V. Cherkassky, J. Anderson, and X. Hu. Support vector machines for temporal classification of block design fmri data. *NeuroImage*, 26: 317–329, 2005.

P.C. Lauterbur. Image formation by induced local interactions: examples employing nuclear magnetic resonance. *Nature*, (242), 1973.

T.H. Le and X. Hu. Potential pitfalls of principal component analysis in fmri. In *Proceedings of the International Society for Magnetic Resonance in Medicine*, page 820, 1995.

E. L. Lehmann and H. J. M D'Abrera. *Nonparametrics: Statistical Methods Based on Ranks*, pages 292, 300, and 323. Prentice-Hall, NJ, 1998.

F. Leymarie and M.D. Levine. Simulating the grassfire transform using an active contour model. *IEEE Transaction PAMI*, 14(1):56–75, 1992.

R. Linsker. Self-organization in a perceptual network. *IEEE Computer*, 21:105–117, 1988.

D.J.C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

J.B.A. Maintz and M.A. Viergever. A survey of medical image registration. *Medical image analysis*, 2:1–36, 1998.

S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), 1989.

D. Marr and E. Hildreth. Theory of edge detection. In *Proc. Roy. Soc. Lond.*, volume B207, pages 187–217, 1980.

S.J. Mason and N.E. Graham. Areas beneath the relative operating characteristics (roc) and relative operating levels (rol) curves: Statistical significance and interpretation. *Q. J. R. Meteorol. Soc*, 30:291–303, 1982.

M.J. McKeown, T-P. Jung, S. Makeig, G.G. Brown, S.S. Kindermann, T-W. Lee, and T.J. Sejnowski. Analysis of fmri data by decomposition into independent spatial components. *Human Brain Mapping*, 6:1–31, 1998.

J. C. Meza. Opt++: An object-oriented class library for nonlinear optimization, 1994.

T. Mitchell, R. Hutchinson, R.S. Niculescu, F. Pereira, and X. Wang. Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004.

P.P. Mitra, D.J. Thomson, S. Ogawa, K. Hu, and K. Ugurbil. Spatio-temporal patterns in fMRI data revealed by principle component analysis and subsequent low pass filtering. In *Proceedings of the International Society for Magnetic Resonance in Medicine*, page 817, 1995.

O. Monga and S. Benayoun. Using partial derivatives of 3d images to extract surface features. *Computer Vision and Image Understanding*, 61(2):171–189, 1995.

J. Munkres. Algorithms for assignment ans transportation problems. *J. Soc. ind.ust. Appl. Math.*, 5:32–38, 1957.

J.C. Nash. *Compact Numerical Methods for Computers: Linear Algebra and Function Minimisation, 2nd ed.* Adam Hilger, Bristol, England, 1990.

J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 1999.

M.J. O'Connell. Search program for significant variables. *Comp. Phys. Comm.*, 8, 1974.

S. Ogawa, D. Tank, R. Menon, J. Ellermann, S. Kim, H. Merkle, and K. Ugurbil. Intrinsic signal changes accompanying sensory stimulation: Functional brain mapping with magnetic resonance imaging. *Proc. Nat. Academy of Sciences USA*, 89:51–55, 1992.

R. L. Ogniewicz and O. Kuebler. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(3):343–359, 1995.

A.V. Oppenheim, A.S. Willsky, and I.T. Young. *Signals and Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1983.

S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

S.M. Polyn, J.D. Cohen, and K.A. Norman. Detecting distributed patterns in an fmri study of free recall. In *Society for Neuroscience conference*, 2004.

E. M. Purcell, H. C. Torrey, and R. V. Pound. Resonance absorption by nuclear magnetic moments in a solid. *Physical Review*, page 37, 1946.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5), 1988.

J.A. Sethian. A fast marching level set method for monotonically advancing fronts. In *Proc. Nat. Acad. Sci.*, volume 93 nr. 4, pages 1591–1595, 1996.

J.A. Sethian. *Level Set Methods and Fast Marching Methods-Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching (presentation slides).

K. Siddiqi, A. Shokoufandeh, S. J. Dickinson, and S. W. Zucker. Shock graphs and shape matching. *Int. J. Comp. Vision*, 1998.

S.L. Smith. Magnetic resonance imaging. *Anal. Chem.*, 57, 1985.

S.M. Smith, P. Bannister, C.F. Beckmann, M. Brady, S. Clare, D. Flitney, P. Hansen, M. Jenkinson, D. Leibovici, B. Ripley, M. Woolrich, , and Y. Zhang. Fsl: New tools for functional and structural brain image analysis. In *Seventh Int. Conf. on Functional Mapping of the Human Brain. NeuroImage*, volume 13, 2001.

H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Proceedings of Shape Modelling and Applications Conference, SMI 2003, Seoul, Korea*, May 2003.

S.P. Tarasov and M.N. Vyalyi. Construction of contour trees in 3d in o(n log n) steps. In *Symposium on Computational Geometry*, pages 68–75, 1998.

A. Telea and J.J. van Wijk. An augmented fast marching method for computing skeletons and centerlines. In *Proc IEEE VisSym*, pages 251–260, 2002.

A. Telea, C. Sminchisescu, and S. J. Dickinson. Optimal inference for hierarchical skeleton abstraction. In *ICPR*, pages 19–22, 2004.

K.R. Thulborn, C. Martin, and J. Voyvodic. fmri using a visually guided saccade paradigm in alzheimer's disease. *AJNR Am J Neuroradiol*, 21:524–531, 2000.

M. Tipping and C. Bishop. Probabilistic principal component analysis, 1997. URL `http://citeseer.ist.psu.edu/tipping99probabilistic.html`.

J.W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.

M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *In Proc. 13th Annu. ACM Sympos. Comput. Geom.*, pages 212–220, 1997.

R. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey, 2000. URL `http://citeseer.ist.psu.edu/veltkamp00contentbased.html`.

M.W. Woolrich, B.D. Ripley, J.M. Brady, and S.M. Smith. Temporal autocorrelation in univariate linear modelling of fmri data. *NeuroImage*, 14(6):1370–1368, 2001.

M.W. Woolrich, T.E.J. Behrens, and S.M. Smith. Constrained linear basis sets for HRF modelling using variational bayes. *NeuroImage*, 21(4):1748–1761, 2004.

Y. Xia. Skeletonization via the realization of the fire front's propagation and extinction in digital binary shapes. *IEEE Transaction PAMI*, 11(10):1076–1086, 1989.

A. Yoshitaka and T. Ichikawa. A survey on content-based retrieval for multimedia data. *IEEE Transactions on Knowledge and Data Engeneering*, 11(1), 1999.

A. Zaimi, C. Hanson, and S.J. Hanson. Event perception of schema-rich and schema-poor video sequences during fmri scanning: Top down versus bottom up processing. In *In Proceedings of the Annual Meeting of the Cognitive Neuroscience Society*, 2004.

C. Zhu, R. H. Byrd, and J. Nocedal. L-bfgs-b: Algorithm 778: L-bfgs-b, fortran routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550 – 560, 1997.

Curriculum Vita

Bing Bai

EDUCATION

Oct. 2007 Ph.D in Computer Science, Rutgers University, New Brunswick, NJ

Jan. 2004 M.S. in Computer Science, Rutgers University, New Brunswick, NJ

Jul. 1997 B.E. in Automatic Control, Tsinghua University, Beijing, China

EXPERIENCE

May. 2007—Sep. 2007 Summer Intern, NEC labs, Princeton, NJ

Sep. 2003—May. 2007 Research Assistant, Department of Computer Science, Rutgers University, New Brunswick, NJ

Jun. 2006—Aug. 2006 Summer Visitor, NSF DIMACS, Piscataway, NJ

Sep. 2000—Jul. 2003 Teaching Assistant, Department of Computer Science, Rutgers University, New Brunswick, NJ

Sep. 1999—May. 2000 Software Engineer, Legend / Lenovo Group, Beijing, China

Sep. 1997—May. 2000 Software Engineer, Tsinghua University, Beijing, China

PUBLICATIONS

"Toward Content-based Indexing and Retrieval of Brain Images", Bing Bai, Paul Kantor, Nicu Cornea and Deborah Silver, in the proceedings of the Recherche d'Information Assistee par Ordinateur '07 (RIAO07), 2007.

"Exploring Interactive Information Retrieval: An Integrated Approach to Interface Design and Interaction Analysis", Gheorghe Muresan and Bing Bai, in the proceedings of the Recherche d'Information Assistee par Ordinateur '07 (RIAO07), 2007.

"A Shape-based Finite Impulse Response Model for Functional Brain Images", Bing Bai and Paul Kantor, in the proceedings of the 4th International Symposium of Biomedical Imaging (ISBI07), 2007.

"A Model for Quantitative Evaluation of an End-to-end Question Answering System", Nina Wacholder, Diane Kelly, Paul Kantor, Robert Rittman, Ying Sun, Bing Bai, Sharon Small, Boris Yamrom and Tomek Strzalkowski, Journal of the American Society for Information Science and Technology, 58(8): 1082-1099, 2007.

"IR Principles for Content-based Indexing and Retrieval of Brain Images", Bing Bai, Paul Kantor, Nicu Cornea and Deborah Silver, in the proceedings of the fifteenth ACM International Conference on Information and Knowledge Management (CIKM06), 2006.

"Automated Judgment of Document Qualities", Kwong-Bor Ng, Paul Kantor, Tomek Strzalkowski, Nina Wacholder, Rong Tang, Bing Bai, Robert Rittman, Peng Song and Ying Sun, Journal of the American Society for Information Science and Technology, 57(9): 1155-1164, 2006.

"The institutional dimension of document quality judgments", Bing Bai, Kwong-Bor Ng, Ying Sun, Paul Kantor and Tomek Strzalkowski, in the proceedings of the 2004 Annual Meeting of American Society for Information Science and Technology, 41(1): 110-118, 2004.

"Designing a realistic evaluation of an end-to-end interactive question answering system", Nina Wacholder, Sharon Small, Bing Bai, Diane Kelly, Sean Ryan, Robert Salkin, Peng Song, Ying Sun, Ting Liu, Paul Kantor and Tomek Strzalkowski, in the proceedings of LREC, 2004.

"Fundamental of TCP/IP and Network Services on Unix", (Chinese, chapters "E-mail" and "Domain Name Service" in book), Dongxing Jiang, Bing Bai, Zhirui Cheng, Qixin Liu and Lin Zhou, Tsinghua University Press, 2003.

"Accounting system for campus network: research and implementation", Dongxing Jiang, Qixin Liu, Bing Bai and Li Qi, Computer Engineering (Chinese), Vol. 26, 2000.