Sensory Mapping, Development and their Applications to Feature and Attention Selection

By

Nan Zhang

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science and Engineering

2006

Abstract

Sensory Mapping, Development and their Applications to Feature and Attention Selection

By

Nan Zhang

Rich literature has been generated on computer vision systems. However, an implementable computational model of immediate vision for general and unknown environments is still illusive. Motivated by the autonomous development process of humans, we are interested in building a robot vision system that automatically develops its visual cognitive skills through realtime interactions with the environment. Developmental vision is required by the demands of general-purpose vision systems for complex human environments.

Based upon the above requirement, in this dissertation we propose a general architecture called Staggered Hierarchical Mapping (SHM) that performs feature derivation for a set of receptive fields and attention selection. The work reported here is motivated by the structure of the early visual pathway. We use several layers of staggered receptive fields to model the needed units of local analysis. From sequentially sensed video frames the proposed algorithm develops a hierarchy of filters, whose out-

puts are maximally uncorrelated within each layer, and contains an increasing scale of receptive fields that range from low to high layers. We also show how this general architecture can be applied to occluded face recognition, which demonstrates a case of attention selection.

Besides this general neural network architecture, we develop several sensory mapping learning rules for deriving feature detectors, including a fast incremental independent component analysis (ICA) method called Lobe Component Analysis (LCA), which derives independent components for many natural cases. A mathematical analysis of the algorithm has been done and which shows the advantages of the LCA method.

We push the research further by investigating the LCA method in a overcomplete setting, which means the number of basis functions is greater than dimension of the observation. A new incremental method is developed to solve the equivalent regression problem with sparse regulization term, i.e. the LASSO regression. We show that the LCA method is a special case when noise is not present. It makes LCA's principal applicable to a large variety of applications, e.g. classification, regression, and feature selection.

To Eric.

ACKNOWLEDGMENTS

While working the last four years on the subject of this thesis I enjoyed the support of my colleagues and other people. Of course, the most direct support came from my supervisors Dr. Juyang Weng, who helped me choose the topic of this dissertation and has been a great source of ideas, comments and encouragement. Without his support, this dissertation would not have completed. I am grateful to Dr. Jain for offering an excellent course on pattern classification and machine intelligence, which I took and learned so much from it. My gratitude also goes to all the professors in my committee, Dr. Ofria, Dr. Henderson, and Dr. Stockman, for providing many valuable critiques and comments on my research topic and for personal support. I would like to acknowledge my special thanks to Dr. Zhang, who took pains to travel across the country to attend my committee meetings and provided research grant for my research project.

I used several pieces of software and code developed by my colleague. Wey-Shiuan Hwang provided the HDR code. Yilu Zhang invested a huge amount of time to help me on the CCIPCA algorithm and code. Shuqing Zeng has jointed force with me to develop many test experiments in this dissertation. Raja Ganjikunta and Martin

Law have shared many valuable ideas with me in the research. Without these, it will take me much longer, if not impossible at all, to finish this work.

Thanks also to the members in both the EI and PRIP Laboratory, past and present, at Michigan State University. I benefited from many discussions with my colleagues. Thanks go to Xiao Huang, Feilong Chen, Zhengping Ji, Wei Tong, Xiaoguang lv, Hong Chen, Yi Chen, Ameet Joshi and Jason Massey. Special thanks to Gil Abramovich, who provided supports throughout and helped me get rid of the occasional doom-thought when working on this dissertation.

I am greatly indebted to my parents for their continuous encouragement and support throughout my life. Thanks also go to my wife, Yue Wang, and my son Eric for their patience and support during the years at Michigan State University.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

The function of current computer vision systems lags far behind the biological visual processing systems. Computer vision systems face many delicate problems, which can be solved easily even by some primary animal vision system. One possible reason is that the brain is built on a much larger scale in the sense of computational units. For example, human brain contains 10^{15} synapses compared to fewer than 10^{8} transistors, plus the more complex function of the neural cells (Kandel et al., 2000).

It is almost impossible to design and manufacture a computer with 10^{15} transistors and make it work. How does nature manage this problem? It is believed that the genome plays a major rule in forming the protein, and thus the tissues and apparatuses. Is the form of synapses completely determined by human genes? The answer is negative. The genome has only approximately $10^{5} genes$. Even if we assume that most of the genes are dedicated to code the synapses, the compression rate is as high

as 1 to 10^{10}. So far, we don't see that much redundancy in the neural synapses. Examination of the neural system revealed that the environment has a large effect on the forms of the neural system. As early as 1970, Blakemore and Cooper (Blakemore and Cooper, 1970) reported that the kittens' visual cortex do not have cells sensitive to edged orientations that they did not observe, if they lived in a controlled environment after birth, where only edges of a certain orientation were present. Recent studies by Sur and coworkers (Sur et al., 1999) have shown that the input signal can fundamentally determine the filters generated under development. They have rewired the visual sensory axon of ferrets to the auditory cortex, and tested the sensory response in it. The auditory cortex shows orientation sensitive cells if it receives visual signals early in life, which can not be found in a natural auditory cortex. Now, it is widely accepted that the neural system is largely formed by its conveyed signals. In engineering terms, this is a data-driven self-organizing system.

Another possible reason that natural vision systems outperform their computer counterpart lies in the selective visual attention mechanism used by the brain. Attention implements an information processing control gate, only allowing a small part of the incoming sensory input to reach short-term memory and visual awareness (Desimone and Duncan, 1995). Instead of trying to fully process the entire sensory input in parallel, nature has devised a serial strategy to achieve near optimal and realtime performance despite the limited computational resources. Selective visual attention has broken down the cluttered visual field into a series of light computational, localized visual analysis.

In this chapter, I will focus on these two problems and review the relevant works.

Our goal is to find the relationship between neuron development and attention selection and explore the possibility that is *learning to pay attention.*

This paper is organized as follows: section 1.2 introduces some background knowledge of the biological visual system. Section 1.3 reviews several models of the early visual processing with the emphasis on modeling the topographic maps of the cortex. Models for visual attention, which is the major concern of this chapter, is shown in section 1.4. In the last section we will discuss how visual system development relates to the learning of visual attention.

1.2 Background of biological visual system

The visual system has the most complex neural circuitry of all the sensory systems. The auditory nerve system contains 30,000 fibers, but the optical nerve contains over one million!

The task of constructing such a system is tremendously complex. Early stage of this process, in which neurons find their proper locations and send axons to roughly the right region of the correct target structure, are somehow known to be independent from the neuron activity, thus could be wired by genes. Later stages of this process, however, depends on the neuron electrical activity to form the accurate synapses structure. What is know about the activity-dependent process of synapse forming is generally consistent with a hypothetical rule that was proposed by Hebb (Hebb, 1949), which is also known as "Hebbian learning rule": Synapses are strengthened if there is a temporal correlation between their presynaptic and postsynaptic patterns

of activity, and weaken otherwise. In a concise phrase, "Neurons that fire together will be wired together".

In this section, I will give a brief introduction to the biological early visual system and review some of the evidence that patterned-activity driven synapse modifications operate in neural development.

1.2.1 Biological visual system

The biological visual system has been studied over several species, including cat, ferret, and monkey. Although their visual systems do contain some structural and functional differences, the main mechanism of the visual processing and development of the circuitry is expected to apply to other species as well.

Fig. 1.1 shows the early visual pathway of the human . The visual processing can be separated into roughly three stages: the *retina*, the *lateral geniculate nucleus* (LGN) and the *primary visual cortex*. Lights projected on the retina evokes the photoreceptors and related cells on the rear surface of the eye. The activity of photoreceptors is encoded by the *retina ganglion cells*. These cells are not evenly distributed. The densest section, which is called *fovea*, is around the center of the visual field.

The visual fields of the left and the right eye have significant overlap, so that the visual field has both binocular and monocular zones. Light from the binocular zone will be projected onto both eye, and light from the monocular zone only appears on one of the retinas.

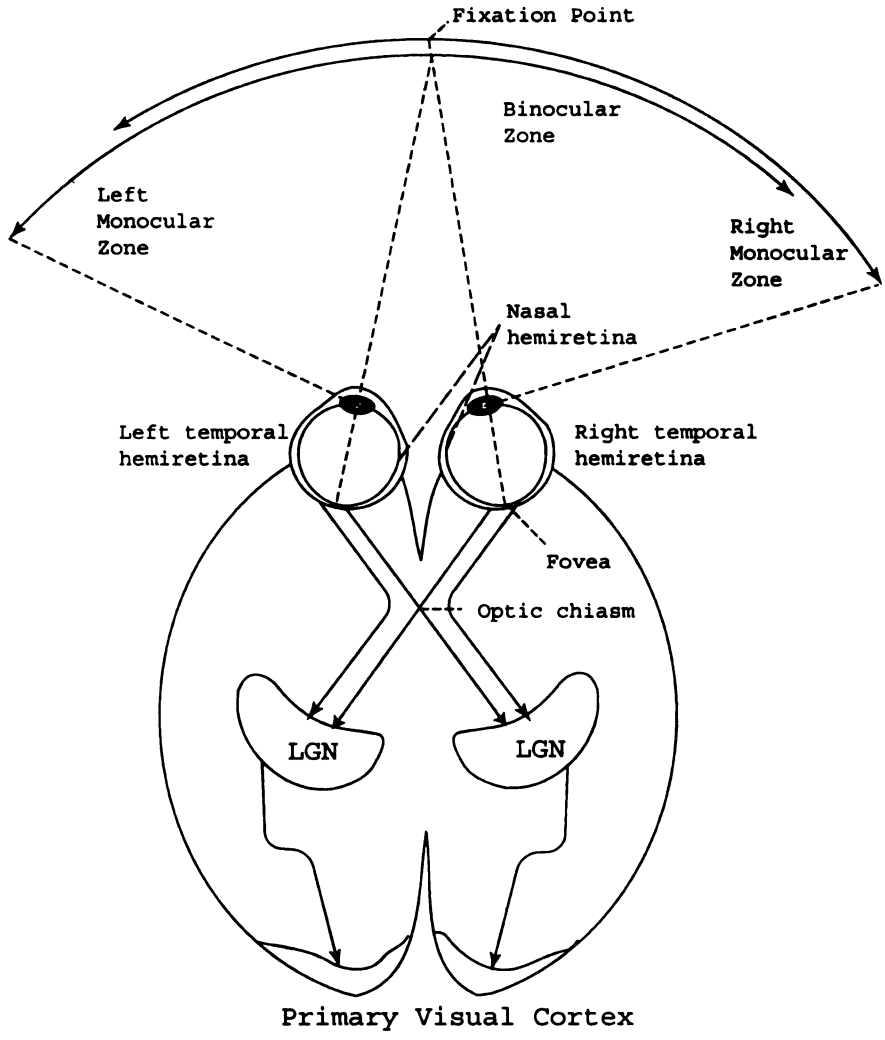


Figure 1.1: Mammal early visual pathway. (Adapted from (Kandel et al., 2000).)

Visual information travels in separated pathways for each half of the visual field. The signal in the left hemifield of both retina will go to the left subcortical area. Similarly the right hemifield only has connections to the right subcortical area. The optic nerves from each eye project to the optic chiasm, where fibers from each eye destined for one or the other side of the brain are sorted out and bundled in the bilateral optic tracts, which project to three major subcortical areas: the *pretectum*,

the *superior colliculus* and the *lateral geniculate nucleus* (LGN). The former two subcortical areas are known to be in charge of saccadic eye movements and pupillary reflexes accordingly, which is out of the topic of this paper. Ninety percent of the retina axons terminate in the lateral geniculate nucleus of the thalamus. The LGN is the main terminus for input to the visual cortex (Kandel et al., 2000). We will focus on the projection to the LGN.

Visual information from different eyes is kept segregated into different neural layers in the LGN. From the LGN, the signal continues to the *primary visual cortex* (V1; also called striate cortex or area 17). V1 is the first cortical area that processes the visual information. The output from V1 goes on many higher layers, including layers underlying some high level cognitive process, like face recognition or top-down attention control.

1.2.2 Early Visual Pathway

The retina, LGN and primary visual cortex constitute the early visual pathway. Although the fundamental function of the early visual pathway is somehow still unclear, many facts already revealed helped us to recognize its functionality.

First, this pathway is the major visual information conducting pathway. Without this pathway visual perception is lost, although some very limited stimulus detection and movement toward objects in the visual field is still possible.

Second, this pathway does not project the original visual field directly into the cerebric cortex. At the photoreceptor level, the representation of the visual field is

much like a image, but significant processing occurs in the subsequent subcortical and early cortical stages, which distorts and recodes the visual information (a review can be found in (Kandel et al., 2000)).

Third, the neural cells in this pathway are organized in a hierarchy and have a variety of receptive field sizes (The receptive field of a cell, by Hubel and Wiesel's definition (Hubel and Wiesel, 1962), is "the region of retina (or visual field) over which one can influence the firing of the cell."). The neurons in the higher layer have larger receptive field than the early neuron. In the same layer, different neurons cover different regions of the visual field. Then the neurons' firing pattern is the representation of the its receptive field. Thus neurons in different layers and at different locations form multi-scale and multi-position representations of the visual field.

Finally, there is not only one single pathway in the early visual processing. As mentioned before, the optical nerves from the left and the right eyes are segregated in this pathway. Furthermore, psychophysical experiments show us that the early visual pathway can also be identified as two parallel pathways: *Magnocellular pathway* (*M channel*) and *Parvocellular pathway* (*P channel*). These pathways start from the Magnocellular Ganglion cells (*M cells*) and Parvocellular Ganglion cells (*P cells*) accordingly, and keep separated until fused in some high cortical layers. *M cells* and *P cells* are proved to be sensitive to different stimuli modalities, e.g. *M cells* are sensitive to luminance contrast and temporal frequency signal. *P cells* are found preferring color contrast and spatial frequency changing. From a later section we will discuss that this is believed to be the main biological evidence of the bottom-up

attention control mechanism.

Fifth, the early visual pathway is largely formed by visual experience. This is obvious as shown by Blakemore and Cooper's newborn kitten experiment (Blakemore and Cooper, 1970) and Sur's optical nerve rewiring experiment (Sur et al., 1999).

The early visual pathway serves an important role in visual information transmission and processing. To study the function of this pathway, one needs to address the processing of individual cells.

1.2.3 Simple and Complex Cells

As we move from retina to primary visual cortex, a fundamental change takes place in the nature of the representation of visual information. In the retina, each ganglion cell effectively represents a small region in the image. The receptive field often receives input from only one cone in the fovea, that effectively samples the image. In frequency terms, each cell has a broad bandwidth that is essentially equal among same type of cells. In the cortex, the situation is strikingly different. Receptive fields are narrow band and oriented, and may differ from one another in their orientation, size and peak frequency. This means that we have gone from a single representation by relatively homogeneous cells, to multiple representations by distinct populations of cells differing in orientation and spatial frequency. Hubel and Wiesel tried to classify the neural cells in the visual cortex by measuring the response properties of visual cortical cells when the subject was exposed to a black screen with only one light spot or light bar visible. The light stimuli is put in different places and the response of the cell is

measured to determine the receptive field property.

They found there are two distinct cell types. The first type of cells has explicit excitation and inhibition regions in their receptive fields. Illumination of part or all of an excitatory region increased the firing of the cell, whereas a light shone in the inhibitory region suppressed the firing. A large spot confined to either area produced a greater change in firing rate than a small spot, indicating summation within either region. This cell group is called simple cells. While receptive fields of simple cells are similar to those of retina and LGN cells in possessing explicit excitatory and inhibitory regions, they differ tremendously in the spatial arrangement of the regions. Fig. 1.2 shows the difference between retina and LGN cells and simple cortical cells. The orientation sensitive arrangement yields simple cells firing profoundly to specific orientated stimuli, thus invoke the hypothesis that simple cells are used as"edge detectors" proposed by Barlow (Barlow, 1972) etc.

The firing patterns of the other group of neurons are far more intricate and elaborate. The receptive fields of these cells are termed "complex". In the experiment, small spots fail to map the excitatory and inhibitory regions. The "on" and "off" firing patterns are triggered by more complex stimulus, such as slits, edges, and dark bars.

The classification of the simple and complex cells is merely the result of the external measurement. They can not be separated anatomically. Be aware of the fact that even the auditory cortex can evolve to have the property of the visual cortex. The difference becomes to lie on the arrangement of its previous layer cells and input signal. Furthermore, the study on a neighborhood of neurons in the visual cortex

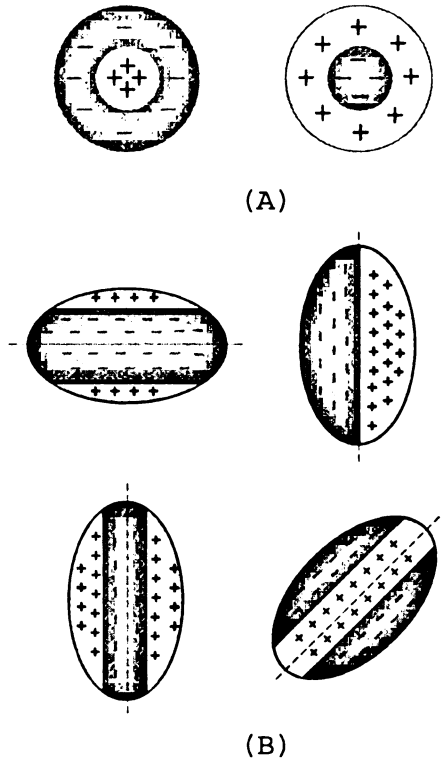


Figure 1.2: The receptive field in primary visual cortex are different and varied than those in the retina and LGN. Areas marked with "+" denote the excitatory region, correspondingly "-" presents inhibitory region. (A) The concentric receptive field of cells found in retina and LGN. (B) Receptive field of simple cells in primary visual cortex.(Adapted from (Hubel and Wiesel, 1962).)

confirms a topographic distribution of cell properties (Adrian, 1941) (Marshall and Talbot, 1942). That is, whether simple or complex cells, cells in a close range tends to fire uniformly to the same stimuli, thus forming a visualized topographic map in the brain.

1.2.4 Topographic Maps in Brain

In all mammals, much of the neocortex consists of orderly representations or maps of neurons that are typically topographic at global levels while being iso-property at

local levels. Two prominent topographic maps are ocular dominance columns and orientation preference maps.

The ocular dominance column represents an orderly arrangement of cells that receive inputs only from the left or right eye and is important for binocular interaction. By injecting radio-labelled amino acids, the ocular dominance columns can be visualized (similar to striped patterns). The neural pathways from the left and the right eye are shown to be largely segregated through the LGN and primary visual pathway.

The orientation preference map is the phenomenon that neurons in the primary visual cortex fire selectively to a specific direction of stimuli. Neurons in the nearby region have similar, but not identical preferences. The preferences repeat continuously at regular interval (approximately 1-2mm) in every direction. Fig. 1.3. shows the orientation preference map of monkey's striate cortex. In some locations, the map forms a pattern like a pinwheel.

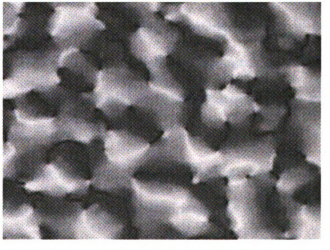


Figure 1.3: The orientation preference map of the macaque monkey's striate cortex, measured by optical imaging technology. Each color presents on preferred direction. From the image we can see the neighbor neurons tend to prefer similar directions, forming the unitary color regions called iso-orientation blobs.

Although the topographic map is currently a prominent research topic in neuroscience, its functioning rule in perception is still controversial (Kass, 1997) (Weinberg, 1997). Weinberg argued that topography could be mere epiphenomenon of minimal intrinsic significance, but he also admitted that it may be useful to examine possible explanations or cortical function. At least, this easy-to-observe phenomenon can be used as a guideline to build the computational model of the early visual pathway, especially when the goal function of this pathway is still unknown. In fact, many neural computation models have attempted to explain the topographic maps with some success. In the following section we will review some computational models that model the topographic maps and simple cells of the early visual pathway.

1.3 Early visual processing models

Visual system development is a complex process. It is difficult to integrate the incoherent experimental results into a profound understanding of how this system is constructed. Computational neural models provide useful ways to recognize functions of the neural system. These could help to validate current theory, and perhaps predict some uncovered facts.

There has been a long history of researchers using quantitative models to explain some observed topographic maps in the brain, such as ocular dominance stripes and orientation preference maps in the early visual pathway. Although limited by the computational power available, some early models did show that orientation selection could be developed. As early as 1973, von der Malsburg showed that orientation selec-

tion neurons could be developed from an oriented image pattern by an unsupervised learning rule (von der Malsburg, 1973). As a pioneering study in this field, von der Malsburg's model had many features that were adopted by many later researchers, such as using two dimension array to present the subcortical and cortical layers, a nearby excitatory and faraway inhibitory mechanism, using the incremental Hebbian learning rule to update the connections.

von der Malsburg's result is not surprising, because the input pattern is oriented. Linsker, however, demonstrated that even with random noise input without pre-applied orientation signal, his model could develop orientation preference maps (Linsker, 1988). In his three consecutive papers (Linsker, 1986), Linsker showed how the "on-off" centralized neuron, the orientation selection neuron, and orientation preference maps can be developed from a simple set of rules (including Hebbian-type modification). Linsker's model gets support from the biological experiment that new born animals possess orientation preference maps even before their eyes open (Crair et al., 1998). This map driven by the internally generated signal is modeled by training with random noise.

Linsker's model motivated several other models that integrate Hebbian learning rule and localized lateral inhibition rule. Like in Miikkulainen et. al.'s RF-LISSOM model (Sirosh and Miikkulainen, 1997) (short for Receptive-Field Lateral Interconnected Synergetically Self-Organizing Map), two layers of neurons model the LGN layer and visual cortex, in which the cortical layer applies a modifiable lateral excitation-inhibition interconnection. Both afferent and lateral connections are adapted by the same Hebbian mechanism in a purely local and unsupervised learning

process. This model can develop the ocular dominance column and orientation preference map. Other similar models that used lateral inhibition and Hebbian learning can be found in (Burger and Lang, 1999) and (Bartsch and van Hemmen, 2001).

Miller (Miller, 1994) used a different approach to make the simulation more practical. He showed that if the visual system is essentially linear, then the sequence of input patterns can be replaced with a simple function representing its long term correlations. His model is essentially Hebbian learning based. Instead of incrementally updating the connection by each input image or pattern, his model integrates the whole sequence of the input in the Hebbian learning equation and comes up with an equivalent form that only demands the correlation function of the image set. This modification significantly improved the convergence speed of the computation. Compared to thousands of images used to training other models, this correlation based model only require tens of the iterations to get a stable result.

In this section we have reviewed several early visual pathway models that explain the development of orientation preference maps. One prominent question should be asked: why is the orientation preference map important? We believe the orientation map, as a low-level feature extraction mechanism, is very important for the attention selection process. As discussed in the next section, the bottom-up attention selection mechanism fuses several feature maps to gain the attention shift position. The orientation map is one of the basic feature maps that contribute to the attention control. Most important of all, the orientation map is developed from the visual experience, which makes attention from learning possible.

1.4 Models for visual attention

The human's brain does not take all the sensory information at one time. Only a subset of all the sensory input can be processed.

Sometimes this selection happens between different sensory modalities. When focusing on reading a book, one could be very insensitive to the surrounding sounds and voices, and when trying to listen to a sound very carefully, one tends to close his eyes to help focus on the auditory sensory input.

In some sensory modes, selection within the modality is also possible. In the visual modality, it is done by suppressing information outside a spatially circumscribed region of the visual field. This is called "focus of attention". In mammals, this attention selection mechanism is implemented either by rapid, saccadic eye movement (also referred as "overt" shift of attention) or so-called "covert" (internal shift of attention, i.e., without moving eyes) shift of visual attention.

In the first case, overt attention can be observed externally as it is orienting the sensor or the highest resolution part of it toward the attended stimuli. This allows the selected information to enter short-term memory and to remain there long enough to reach conscious and cognitive levels of representation. Most mammals' eye systems have some anatomical specification to accommodate this "spotlight" mechanism as well. In many species' retina, the photoreceptors are not distributed in a even manner. They contain a specialized region around the center of the visual field with a much higher cell density than at the peripheral region. This is called "fovea" in primates, "central area" in cats and "visual steak" in rabbits.

In contrast, covert attention is not directly observable but takes place by filtering or assigning the resource. Covert attention selection usually precedes the overt attention mechanism (Ditterich et al., 2000). Like in a task of fast visual searching for an object, one looks at the scene without a specified fixation, then overt attention filters irrelevant distraction. After a few candidate positions are selected, saccadic eye movement takes place for further visual analysis.

Although the overt and covert visual attention are modelled in similar ways, they have different costs and effects. Moving sensors takes a longer time than the internal attention selection. Further more, the overt attention selection requires shutting down the sensory input during the saccade. But in some realtime active-vision system, saccadic sensory movement is necessary to prevent the attended object from moving out of the visual field and take advantage of the highest sensory resolution.

There are at least two advantages that a visual attention mechanism can bring about. First the amount of data to be processed has been reduced, resulting a lower computational resource demand. Thus, it enables further information processing. Second, the distracting information is suppressed in this procedure, so that relevant information may take more consideration. For example, in a typical object classification system, it is very difficult to deal with samples that come with a large variety of backgrounds. With the help of visual attention selection, the background information is somehow suppressed, thus only the foreground object will influence the performance of the system. In this aspect, attention is a central part of the system, because it selects the information on which the system activities are based.

Selective visual attention has been researched in psychology, physiology, and neu-

roscience for decades (James, 1981) (Kandel et al., 2000) (Treisman and Gelade, 1980), but the majority of the research focuses on empirical aspects, measuring the phenomenon, describing the result, and giving at most qualitative models of the attention control. Computational models of selective attention have drawn research interest only in the last decade. Although largely based on the qualitative models, the computational models of attention casts a light on solving some difficult computer vision problems, and in turn as a quantitative approach to help understanding the biological visual attention mechanism.

In the following section we first give some background knowledge of the attention mechanism in the brain, and then review several computational models of selective visual attention. It is almost impossible to cover all the related work in this area. Some good reviews can also be found in this literature (Itti and Koch, 2001) (Desimone and Duncan, 1995).

1.4.1 Attention mechanism in the brain

The control of selective visual attention involves several brain regions, including the early visual pathway and the prefrontal cortex. Cooperation between two streams of control signal is believed necessary to achieve attention selection. The *where* stream will find where to attend to, primarily controlled by the dorsal visual processing pathway, which comprises cortical areas in the posterior parietal cortex. The *what* stream controlled by the ventral visual processing pathway, involving cortical areas in the inferior temporal cortex, is primarily in charge of localized object recognition. The

result of the two streams could affect each other. For example the recognition result from the *what* stream can bias the next attention shift, make it to some predicted focal attention area where it may contain next interesting object, and vice versa.

Fig. 1.4. shows the brain areas that participate in the control of visual attention.

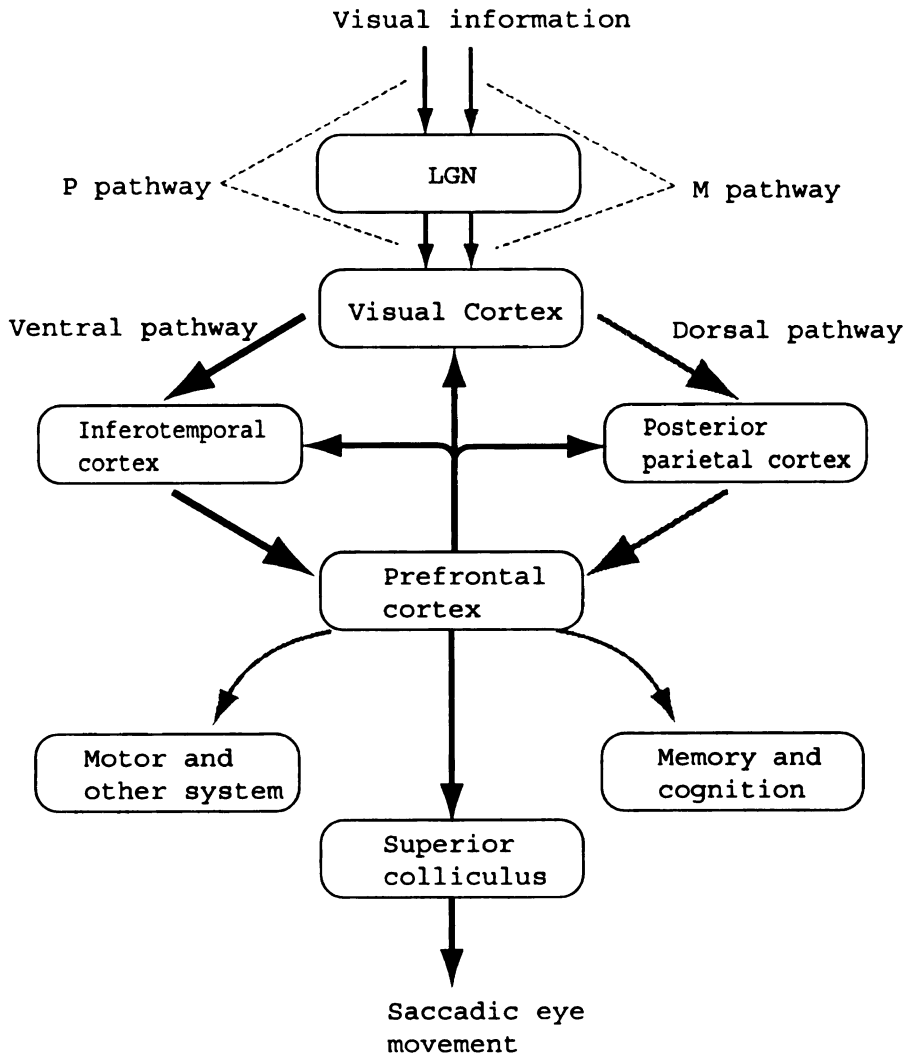


Figure 1.4: Brain areas that involve in attention control. This diagram only shows the control flow. The spatial layout of the pathways can be found in (Kandel et al., 2000). (Adapted from (Itti and Koch, 2001) and (Kandel et al., 2000))

Visual information enters the retina and is processed by two kinds of ganglion cells: the magnocellular ganglion cells (M cells) and the parvocellular ganglion cells

Stimulus feature	M cells	P cells
Color contrast	No	Yes
Luminance contrast	Higher	Lower
Spatial frequency	Lower	Higher
temporal frequency	Higher	Lower

Table 1.1: Differences in the sensitivity of M and P cells to stimulus features. (Adapted from (Kandel et al., 2000))

(P cells). The information that is conveyed by M and P cells is kept segregated in the LGN and projected to separate layers of the primary visual cortex. Thus, it has led to the view that the separate sequence of retinal ganglion, LGN, and visual cortical cells can be regarded as two parallel pathways, referred to as the M and P pathways. As indicated in Table 1.1, there are striking differences between cells in the M and P pathways. By selectively removing one or the other pathway, experiments show that the M and P pathway make different contributions to perception. The P pathway is critical for color vision and for vision that requires high spatial and low temporal resolution (e.g. static color images). The M pathway contributes most to vision requiring low spatial and high temporal resolution (e.g. motions).

Based on anatomical and functional evidence, Ungerleider and Mishkin argued that visual processing centers in the cerebral cortex are also believed to be organized in two pathways: the dorsal processing pathway to the posterior parietal cortex and the ventral pathway to the inferotemporal cortex (Ungerleider and Mishkin, 1980). Anatomical evidence also shows that the dorsal pathway is the successive visual pathway corresponding to the M pathway in the early visual processing, and the ventral pathway corresponds to the P pathway accordingly. Thus it is natural and reasonable to assume the dorsal and ventral pathways implement some high level perception

combination of the information that the M and P pathways convey.

The dorsal pathway is believed to process a subset of high level perceptions, including motion and depth. These perceptions are very likely derived from the M pathway information(motion and depth perception from the low spatial high temporal feature). The neurons in this system are relatively insensitive to color and poor at analyzing stationary objects. Therefore, the dorsal pathway is more concerned with *where* objects are rather than *what* they are.

The ventral pathway deals with color and form perception. The color and the form perception are derived from P pathway features(color and form perception from color contrast and high spatial low temporal feature), because a great deal of information about shape and form is derived from colors, edges, and borders. This system is capable of high resolution, which is important for seeing stationary objects in detail. Thus the ventral pathway in more concerned with *what* is seen.

The dorsal pathway (*where* stream) and the ventral pathway (*what* stream) end in different regions of the prefrontal cortex that specialize, respectively, in visual spatial working memory and object recognition working memory.

1.4.2 Bottom-up attention selection

How is information about color, form, depth, and motion, which is conveyed by different neural pathways, combined together and translated into a comprehensive perception? When we see a red moving car, we do not see separated features. We combined different properties, color (red), shape (car), and motion (linear moving),

into one perception. It is impossible to develop different specialized neural cells to detect all these combinations, thus there must be fusion of these pathways somewhere in a higher level, which is also called a *binding mechanism.*

From a computational aspect, this fusion of low level features resulting a high level attention selection is also referred as the bottom-up attention selection mechanism.

One of the earliest theoretical models that addresses this problem is in the psychophysical literature. Treisman et al. has proposed that different features are derived from different *feature maps* in different brain regions (Treisman and Gelade, 1980). To solve the binding problem, the brain dedicates some processing as a *master map* to combine the code of the feature map. Fig. 1.5 shows the outline of Treisman's model. Later, more detailed and neurally plausible models for visual attention selection were proposed (Koch and Ullman, 1985) (Eriksen and Yeh, 1985) (Olshausen et al., 1993). These models are still qualitative models.

The most prominent model of visual attention was proposed by Koch and Ullman in 1985 (Koch and Ullman, 1985). The model includes following elements:

1. It has a process called early representation, in which a number of elementary features, such as color, orientation, direction of movement, disparity etc. are represented in parallel in different topographical maps. This corresponds to the feature map of Treisman's model.

2. It has a selective mapping from these representations into a central representation which at any instant contains only the properties of a single visual location, the selected location. This is corresponding to the master map.

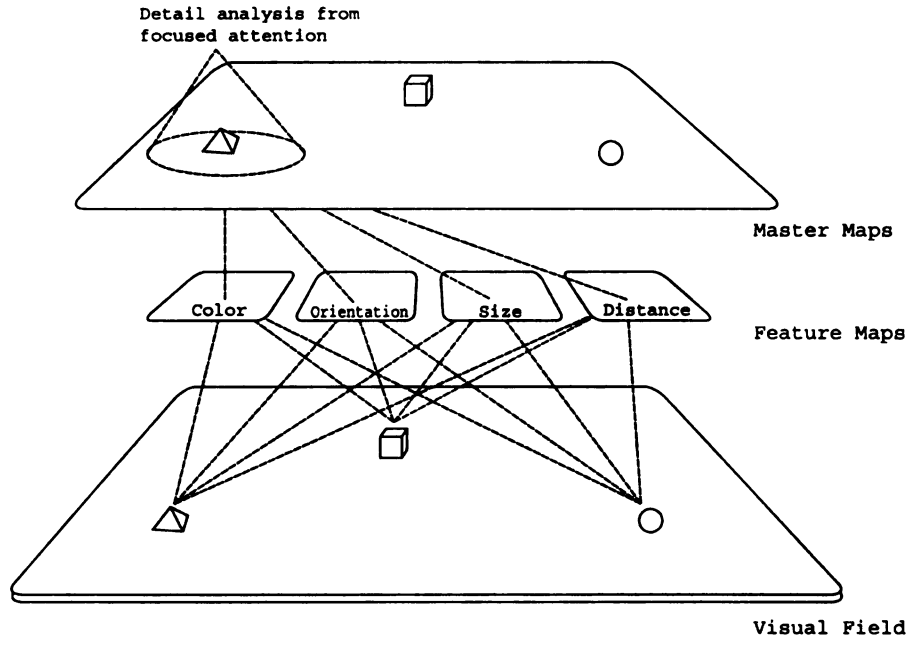


Figure 1.5: Bottom-up attention selection model. The elementary properties (such as color, orientation, size, and depth) of the visual field are extracted by separate parallel pathways, each of which generate a *feature map*. Selected features are fused into a *master map*, which is a representation of those features that have the most saliency. Selective attention happens after the features have been associated in a small region of the master map. Adapted from (Treisman, 1986).

3. It has certain kinds of rules that determine which location is mapped to the central representation. This is implemented by a Winner-Take-All network.

4. It has a mechanism called inhibition of return, which will suppress current attended location and enable the shifting to the next salient location.

Itti and Koch et al. proposed a pure bottom-up attention control model (Itti et al., 1998) (Itti et al., 2001). They integrated color, intensity, and orientation as basic features. A Gaussian pyramid was created to extract intensity information in six scales. Similarly, Gaussian pyramids was constructed to extract features from red-green opponency channel, blue-yellow opponency channel. Four orientation-selective pyramids

were also created using Gabor filtering at 0, 45, 90, and 135 degree in six different scales. A total of 42 feature maps was thus created: 6 for intensity, 12 for color, and 24 for orientation. Then, they fused feature maps into a so-called saliency map. The task of the saliency map is to compute a scalar quantity representing the salience at every position in the visual field. In their combination scheme, they promoted those feature maps with a few strong peak and suppressed those with many comparable peaks or just evenly distributed. This combination strategy outperforms the regularly normalization method, and also keeps biological plausibility (Sillito et al., 1995). After normalization, the feature maps for intensity, color, and orientation are summed across scales into three separate conspicuity maps, each for different modality. Searching for the most salient position is done by a winner-take-all (WTA) network with a global lateral inhibition (Koch and Ullman, 1985). It has long been known that lateral inhibition in neural network can lead to a WTA competition, so that only a single neuron is active at a steady state. To avoid reselecting the first attention position repeatedly, they have used a mechanism to inhibit the recently selected object and shift to another salient position. In psychophysics, this process is called *inhibition of return.*

Backer et al. applied similar strategy to an active-vision system, called NAVIS (Neural Active VISion), which emphasizes the visual attention selection in a dynamic visual scene (Backer et al., 2001). Instead of directly using some low level features like orientation and intensity, they accommodate additional processing to find middle level features to build the feature map, such as symmetry and eccentricity. Fusion of conspicuity maps has been done by a so-called Dynamic Neural Fields, proposed by

Amari (Amari, 1977), which, in fact, is still a lateral inhibition network. Due to the nature of a dynamic scene, the inhibition of return processes is somehow more difficult than visual searching in a stationary scene. To name one, the object that was first picked up and then inhibited could be moving, resulting a possible selection of the next attention shift. In case of a gaze shift, the positions of the saliency before and after shift should be corresponding. In Backer's model, these problems are addressed by specific approaches.

Many of the bottom-up attention selection models share similar ideas. The major differences lie in the variety of features, different strategy of fusing the feature map, and the rules that find the most salient location.

Although the bottom-up attention model may explain the first few hundreds of milliseconds after presented with a new scene, obviously a more complete model of attentional control must include a top-down, volitional biasing shift as well.

1.4.3 Top-down attention selection

Top-down attention selection has involved much complex perception procedures. One obvious phenomenon is that we are able to move our eye voluntarily towards any position, no matter how inconspicuous it is. It is believed to include object recognition, understanding of the scene, and even some planning. The precise mechanism by which a voluntary shift is elicited remains elusive, although several researches have inspected the possible brain regions that are involved (Hopfinger et al., 2000).

As to our knowledge, there are no computational model can explicitly solve this

problem right now, although some models, combined with bottom-up attention selection, tried to accommodate this mechanism.

Tsotsos et al. (Tsotsos et al., 1995) implemented attention selection using the combination of a feed forward bottom-up feature extraction structure and a top-down selective tuning (of these features) extraction mechanism. First, the representation in the net is calculated, based on a feed forward activation for low level to high level. Then the target of attention is selected. The location is then propagated back through the feature extraction hierarchy. In the top-down process, the WTA process is reduced to the units connected with the winner of the previous layer. In the lowest layer, this procedure results in an accumulation of units corresponding to the most conspicuous region in the image.

Although Tsotsos's model includes some top-down control features, it is not usually considered as a true top-down attention selection model. The model's top-down selective tuning only helps select the conspicuous location. It does not know what it is attending to. A successful top-down attention system should accommodate the object recognition capability in order to implement the "what" stream in the brain attention control pathway.

One of the earliest models that combines object recognition and attention is MORSEL (Pashler, 1996), in which the attention is shown to help recognition. This model is designed to recognize words. Without attention, the neighborhood of word could conflict to yield ambiguity and confusion of the recognition. The addition of top-down attention control allow the system to focus one word at a time.

Schill and colleagues proposed a model combining a bottom-up feature extrac-

tion component and a top-down, knowledge based recognition component (Schill et al., 2001). The first component acts like the aforementioned bottom-up attention selection model. The second component contains a hierarchical knowledge tree, which represents object classes by defining several critical point and corresponding eye movement commands that maximize the information gain. This permits the model to discriminate between the candidate object classes.

1.4.4 Summary

In this section, we have reviewed the brain mechanism of attention control and two basic attention selection strategies: bottom-up and top-down. Although bottom-up attention selection is more mature and has been implemented in several successful computational models, a successful focal attention selection model relies on a good combination of the two strategies.

1.5 Discussion

In this chapter, we have reviewed several early visual pathway models and attention control models. These two problems, at a first approximation, are different and unrelated. But after reviewing several related works, we find it is possible to put them in a single theoretical framework.

The topographic map in the early visual pathway may serve as a feature or saliency map in the bottom-up attention control process. The lateral inhibition mechanism in the model of orientation preference map is also adopted by the saliency map fusion of

the attention control models. This leads to a possible uniform solution that addresses attention from learning.

The following chapters will focus on the model of feature maps and attention selection models. Two learning mechanisms and a general vision architecture will be presented and discussed.

Chapter 2

Staggered Hierarchical Mapping for Attentive Perception: A Developmental Model

In this chapter we propose a general architecture that can accommodate attention selection and development of a vision system. The work reported here is motivated by the structure of the early visual pathway. We used several layers of staggered receptive fields to model the pattern of positioning of the processing cells of the early visual pathway. From sequentially arriving video frames the proposed algorithm develops a hierarchy of filter banks, whose outputs are uncorrelated within each layer, but with an increasing scale of receptive fields that range from low to high layers. We have also shown how this general architecture can be applied to occluded face recognition, which demonstrates a case of attention selection.

2.1 Introduction

There has been rich literature about the computer vision system. However, an implementable computer vision model for a general purpose attention guided recognition for unknown environments is still illusive. The appearance-based approach, e.g. eigenface (Kirby and Sirovich, 1990; Turk and Pentland, 1991), has opened the possibility of dealing with unknown, uncontrolled, and complex real world scenes because of its capability of deriving features from any vector distribution.

The appearance-based approach can learn any type of invariance exhibited in training samples, but it requires preregistered objects because it lacks of attention selection capability. Current techniques of the appearance-based approach require that the object to be recognized being registered in a predetermined position and size. This is either provided manually by a human or by a separate object detector. However, in many scenarios this is not applicable.

Along the line of the appearance-based approaches, the developmental vision architecture, proposed here, is motivated by the developmental process of biological vision systems. The idea of developmental vision is consistent with the idea of purposive active vision (Aloimonos, 1992), because the developmental process is purposive and active and so is the performance process. If the system is exposed to a wide variety of environments and has learned many tasks in the developmental process, the resulting system is large, complex and it "acts like" a general-purpose system. If the system is trained in a controlled setting for a single task, the resulting system is small, lean and it "acts like" a special purpose system.

An important factor in such a general architecture is the attention mechanism. An object must be recognized at any position and size after it is projected onto the retina, without turning the eyes. Attention selection is essential for suppressing irrelevant information. Otherwise the image vector that includes the object of interest and the background is a totally new vector that does not have a good match from the learned experience. To learn an object, active attention must be applied not only in the performance session, but also in the learning session.

Many studies have proposed hierarchical models for vision and pattern recognition systems, but few have addressed attention selection as a major goal. Fukushima et al., proposed a neural network called "Neocognitron" (Fukushima, 1980). The Neocognitron is a multilayered network consisting of cascaded connections of many layers of cells. The information of the stimulus pattern given to the input layer is processed step-by-step through stages of the network. The synapses between the neurons are updated by a supervised or unsupervised learning method, but the features retained are hand-picked by humans. Weng et al., proposed a dynamic neural network model called "Cresceptron" (Weng et al., 1997), which can automatically grow a hierarchy of maps directly from image inputs by a learning-with-teacher process. The network grows by creating new neurons, connections and an architecture which memorize new image structures and context as they are detected. Although these studies address the global structure of sensory mapping, the efficiency and completeness of the generated feature representation are problematic, since those methods do not explicitly use statistical properties of the signals in the filter generation.

Other studies concentrate on the subproblem of feature derivation from statistical

properties of input images. They include the entropy reduction method (Atick and Redlich, 1993), the sparse coding method (Olshaushen and Field, 1997), and the independent component method (Bell and Sejnowski, 1997; Hyvärinen and Hoyer, 2000). The above studies applied the principal component analysis (PCA) or the independent component analysis (ICA) to a single size, fixed receptive field. In order to deal with compounding problems of multiple receptive fields, a general objective recognizer must be able to recognize a pattern at different positions and with different sizes. In our proposed architecture, we use a hierarchical structure with staggered receptive fields inside each layer.

Some researchers (Linsker, 1986; Sirosh and Miikkulainen, 1997) used a hierarchical network to simulate the effect of a population of neurons and to explain some biologically observed phenomenons, such as orientation preference maps. These studies did not address the representation issues nor how to use the representation for later classification and recognition. Attention was not a subject of concern in these studies. We believe that a major purpose of the early visual pathway is to provide a coupled solution to two mutually dependent problems: (a) provide representation of input using the current attention selection (suppressing unattended area), to the later recognition and attention signal generator; (b) execute the attention generation by the later recognition and attention signal generator.

These two issues combined together accomplish both the bottom-up and top-down information flow.

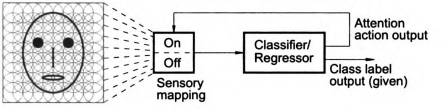
This raises a series of new issues such as how receptive fields are organized, the criteria of feature representation, how attention is applied, and how the output of the

sensory mapping is used.

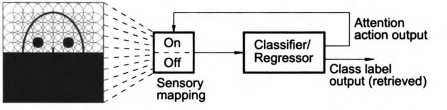
Another essential requirement for a sensory mapping developmental algorithm is that the computational method must be incremental without estimating the covariance matrix directly from online real-time experience. Further, the computation must be quick to respond in real-time. For sensory mapping, this means that the current sensory mapping must be updated within the time interval of every sensory frame. This indicates the challenges in the design of a real-time developmental program.

This chapter proposes architecture of hierarchical sensory mapping, called Staggered Hierarchical Mapping (SHM), for attention selection and attentive-based recognition and its developmental algorithm for active vision systems. Fig. 2.1 illustrates a case of how the sensory mapping is used. The algorithm develops filters for a family of receptive fields. The responses of all filters are available for further analysis by the following classifier/mapping function. Therefore the proposed sensory mapping develops a hierarchical representation for a large set of receptive fields, but does not complete vision by itself. The following classifier/mapping function is needed to work in tandem. The autonomous development of cells in the visual sensory mapping is performed by the CCIPCA (Weng et al., 2003) method, which addresses the convergence problem with existing incremental PCA methods. However, the other filter derivation methods, in addition to PCA, can also be used, e.g. Independent Component Analysis (Olshaushen and Field, 1996; Bell and Sejnowski, 1997) and Slow Component Analysis (Wiskott and Sejnowski, 2002) by the proposed architecture. In this chapter, we used CCIPCA as an example.

An important issue for developing a sensory mapping is the completeness of repre-



(a) Learning session



(b) Performance session

Figure 2.1: Recognition under occlusion through features extracted using a sensory mapping architecture. (a) In the learning session the sensory mapping learns the features of the upper face (indicated by "On") through active attention to the upper face, (b) In the performance session the lower part of the face is occluded and attention focused on the upper face allows the sensory mapping to deliver features for the non-occluded part of the face ("On" part of the block), which are fed to the classifier for successful recall. In general, a sensory mapping enables suppression of unattended receptive fields, but provides signals from attended receptive fields for generating attention control signals by the following classifier/mapping function.

sentation. Since the environment is unknown at the time when designing the system, the programmer designs the mechanism, through which the actual representation (including network architecture, neural weights and responses) is developed from the sensory experience. However, if the mechanism does not allow the derivation of representation that is essential for any particular environment, the resulting mapping is deficient in performance. If the mechanism enables the representation to be complete, in the sense of no loss of information, then the resulting representation is able to handle any environment that an intelligent agent runs into. The completeness for a single receptive field is clear, due to the reconstruction of PCA. However, the reconstruction issue for several staggered receptive fields is illusive. We address the completeness of the resulting SHM mapping by reconstructing the original images from the generated representation.

The subsequent organization of this chapter is as follows. Section 2.2 explains the structure of the proposed SHM model. Section 2.3 describes the developmental method. In Section 2.4, we analyze the computational complexity of this method. Section 2.5 discusses the completeness of representation in terms of reconstruction. Section 2.6 depicts the proposed model as an attention effector. In Section 2.7 we show the results of some of the experiments with SHM. Section 2.8 presents a classification system in which the SHM is used for attention selection. Section 2.10 provides concluding remarks.

2.2 Staggered Hierarchical Mapping

We assume each line of sensory input can be digitized into a real number. It is represented by Euclidian n-space $\mathcal{R}^n$.

For the case of visual sensory input, we can assume the input units are organized in a 2-D plane with fixed r rows and c columns. So all the visual sensory inputs form an $\mathcal{R}^{r\times c}$ space, where $\mathcal{R}$ is the real space. Thus we have

$$\mathbf{X}(t) = [\mathbf{x}_{ij}(t)] = \begin{bmatrix} x_{11}(t) & x_{12}(t) & \dots & x_{1c}(t) \\ x_{21}(t) & x_{22}(t) & \dots & x_{2c}(t) \\ \dots & \dots & \dots & \dots \\ x_{r1}(t) & x_{r2}(t) & \dots & x_{rc}(t) \end{bmatrix} \tag{2.1}$$

We also call a visual sensory input an image. We will use this two terms interchangeably in this chapter.

Analysis on this matrix space $\mathcal{R}^{r\times c}$ is difficult, because it introduces some unnecessary complexity. Therefore, the vector space model is used to represent images. Under this model, each image is modeled as a single vector and the collection of images is modeled as a single data matrix, where each column of the matrix corresponds to an image and each row corresponds to a feature dimension. Thus, we introduce a vectorization operator to convert the 2-D matrix representation to the vector space presentation.

Definition 2.2.1 *Let* $\mathbf{A} \in \mathcal{R}^{r\times c}$ *and* a_j *its* j*-th column, then* $\operatorname{vec}\mathbf{A}$ *is an* $rc \times 1$

vector,

$$\text{vec}\,\mathbf{A} = \left[\mathbf{a}_1^\top \mathbf{a}_2^\top ... \mathbf{a}_c^\top\right]^\top .$$

Then it is natural to get the vector space representation $\mathbf{I}$ of an image $\mathbf{X}$, i.e. $\mathbf{I} = \text{vec}\,\mathbf{X}$.

Without knowing the original row and column dimension, the vector representation lose information of the $\mathcal{R}^{r \times c}$ space. So we have an important assumption: the statistical property of the image is stationary. Basically, the stationary assumption states that the statistical property of an image space is independent of positions of the image. With the stationary assumption, the statistical property of the image space can be captured by the covariance matrix and higher order tensors of the image vector space.

In SHM, a filter is modeled by a column vector $\mathbf{w}_i$, and $\mathbf{w}_i$ has the same dimension as the image vector representation $\mathbf{I}$. All filters that have connections to the image then form a matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ...\mathbf{w}_n]$, where n is number of filters.

The responses of a set of filters $\mathbf{R} = f(\mathbf{W}^\top \mathbf{I})$, where $\top$ is the transpose operator and $f(\cdot)$ is a non-linear function. In the proposed architecture, we did not consider the non-linear function $f(\cdot)$, thus the model is a linear one.

When a family of receptive fields are considered, we have observed that the Principal Component Analysis is a suitable mechanism for filter development. It gives a set of filters that minimize the mean square error between the input and reconstructed space, giving a fixed dimensionality under a linear projection. The response from the PCA filters are mutually uncorrelated. This increases the efficiency of the sensory

coding. Furthermore, incremental PCA that adopts a Hebbian learning mechanism has some support from the biological model of the cortex. It has been shown (Oja, 1982; Sanger, 1989; Rubner and Schulten, 1990) that artificial neurons updated by the Hebbian rule, while being inhibited by nearby neurons (lateral inhibition), produce synaptic weights that are principal components produced by PCA.

However, the conventional PCA also has its limitations. It is only applied to the full input vector, and it is not capable of extracting local features in the input space. We shall design an architecture, in which the responses are computed from a family of localized receptive fields.

A stack of filters developed from a single receptive field is shown in Fig. 2.2(a). We have two problems: First, the spatial resolution is low if the receptive fields do not overlap; Second, if they overlap, the computational cost is high. In the proposed staggered scheme, we trade off spatial resolution with computational cost as shown in Fig. 2.2(b). In the proposed architecture, filters are not stacked to cover the same receptive fields. Instead, the receptive fields of filters are staggered and cover different regions. In practice, this means each filter only gets input from a local region of the input. However, for the convenience of analysis, we shall still model filters to cover the entire input range. Therefore, the matrix $\mathbf{W}$ is usually a sparse matrix, since filters are not connecting to all the inputs, and the connection, or synapses, are localized. This definition intertangled with the issue of filter development makes the analysis much more difficult. So, we introduce a receptive field matrix to restrict the connection to the local area, and thus set no constraints to the filter matrix $\mathbf{W}$.

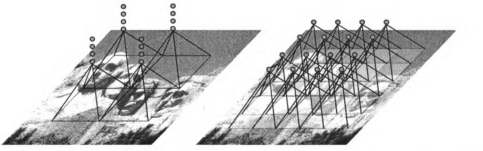


Figure 2.2: A comparison of the neural cylinder (a) and the staggered receptive fields (b). In (a) a group of filters share the same receptive fields resulting in a low spatial resolution. In (b) each filter is centered at a different position resulting in a higher density of spatial sampling.

Definition 2.2.2 *The receptive field* $\mathbf{D}_i$ *of the filter* $\mathbf{w}_i$ *is a diagonal matrix*

$$\mathbf{D}_i = \begin{bmatrix} d_{i,1} & 0 & \dots & 0 \\ 0 & d_{i,2} & \dots & 0 \\ \dots & \dots & \dots & 0 \\ 0 & 0 & \dots & d_{i,K} \end{bmatrix},$$

with $d_{i,j} = 0, 1$ *and* $K = r \times c$ *is the dimension of the image vector space*

The single filter response can be written as $r_i = f((\mathbf{D}_i\mathbf{w}_i)^\top \mathbf{I})$. $\mathbf{D}_i\mathbf{w}_i$ defines a filter with a receptive field.

This definition did not constrain the position of the receptive fields, so the synapses can be scattered in the entire image. But in practical, the receptive field of the filter is usually localized to a small field in the image. However, this topological information is discarded during the vectorization process. So, do we lose important information? The answer to this question is in two fold. First, we do lose the exact topological relationship between the filters. However the correlation between filter pairs can be reserved in the covariance matrix and higher order correlations can

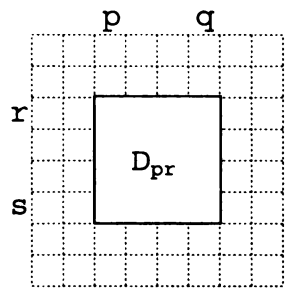


Figure 2.3: Localized receptive field $\mathbf{D}_{pr}$ with size 4.

be found in the higher order tensors. Also the signal is natural image, instead of white noise, so there exists high correlation among the neighborhood input areas. In this sense, the topological structures are encoded in the input signal. A data-driven learning mechanism can find this high order correlation, which thus determine the receptive field. It is worth noting that the introducing of the receptive field matrix $\mathbf{D}$ is merely for the convenience of mathematical analysis. In the actual implementation the weight vector is the product of $\mathbf{D}_i\mathbf{w}_i$, where zero connections are all omitted to save computational resources.

A localized receptive field is defined by its starting row p and starting column r and size z. For the simplicity of the model, we usually assume every localized receptive fields in the same layer has the same size z. From now on, if not otherwise specified, we will always assume this condition. Fig. 2.3 illustrate a localized receptive field $\mathbf{D}_{pr}$ with size 4.

Filters are partitioned into non-intersected groups $\mathcal{G}_i, i = 1, 2...$, which are called eigen-groups. Within an eigen-group $\mathcal{G}_i$, the receptive fields of the filters have no overlap nor gap between them. Therefore, the receptive fields of filters in the same group are paved seamlessly to cover the entire image, i.e. for any $k \neq l$ and $\mathbf{D}_k, \mathbf{D}_l \in$

$\mathcal{G}_i$, we have

$$\mathbf{D}_k\mathbf{D}_l = \Theta, \tag{2.2}$$

where Θ is zero matrix, and

$$\sum_{\mathbf{D}_k \in \mathcal{G}_i} \mathbf{D}_k = \mathbf{I}, \tag{2.3}$$

where the summation is element-wised matrix addition and $\mathbf{I}$ is the identity matrix. In SHM model, filters in the same eigen-group compute the PCA of the same order. Basically, if the size of receptive field is $n \times n$, then there could be maximally $n \times n$ different eigen-groups. None of the eigen-groups share same receptive field, i.e. $\forall \mathbf{D}_i \in \mathcal{G}_k, \mathbf{D}_j \in \mathcal{G}_l$, where $k \neq l$, $\mathbf{D}_i \neq \mathbf{D}_j$. Then we can define distance between eigen-groups as

$$d = \min\{|p - r|, |q - s|\} | \forall \mathbf{D}_{pq} \in \mathcal{G}_k, \forall \mathbf{D}_{rs} \in \mathcal{G}_l, k \neq l.$$

We call the minimum value of d "inter-filter distance", or "inter-neuron distance". Typically, the structure of SHM is homogeneous, so the inter-filter distance is the distance between any adjacent filters.

A randomly predefined order is given to all eigen-groups, for example shown by a number in Fig. 2.4. The numbers determine the order of the eigenvector computed by the eigen-group, and the position of the numbers determines the relative position of the eigen-group. After each input image coming in, the filters in the lower order eigen-group is updated first, and their responses are generated. The filter updating rule will be discussed in section 2.3. Here, we first take a look at what we called residue image processing between the computation of consecutive eigen-groups. Suppose we

15	4	1	6
3	8	9	10
12	13	7	14
2	5	0	11

Figure 2.4: Computational order of eigen-groups. There are 16 eigen groups. The numbers in the table denote the order of updating using residual image. The positioning of the numbers represents the relative position of the eigen-groups.

have finished eigen-group $\mathcal{G}_k$. The response of the eigen-group is $r_i = (\mathbf{D}_i\mathbf{w}_i)^\top \mathbf{I}_{i-1}$, where $\mathbf{D}_i \in \mathcal{G}_k$, $\mathbf{I}_{i-1}$ is the residue image from the last eigen-group and $\mathbf{I}_0 = \mathbf{I}$. The residue image after this computation is defined by

$$\mathbf{I}_k = \mathbf{I}_{k-1} - \sum_{\mathbf{D}_i \in \mathcal{G}_k} (\mathbf{D}_i\mathbf{w}_i)^\top \mathbf{I}_{k-1} (\mathbf{D}_i\mathbf{w}_i) \tag{2.4}$$

The residue image can be considered as the remaining variances that are not captured by the current eigen-group, thus serves as the input of the next eigen-group.

Beside the feed forward processing, filters also have interactions in the same layer. If we assume the lateral interaction is linear, then we can use another matrix to model it.

Definition 2.2.3 *The lateral interaction matrix* $\mathbf{L}$ *is an* $n \times n$ *matrix,*

$$\mathbf{L} = [l_{ij}]_{n \times n}$$

The element l_{ij} *determines the inhibition or excitation synapse from filter* i *to filter* j.

If $\mathbf{L}$ is a symmetrical matrix, then the lateral interaction is symmetrical, asymmetrical otherwise. The responses are affected by both $\mathbf{W}$, $\mathbf{D}$ and $\mathbf{L}$, i.e. $\mathbf{r}_i = (\mathbf{L}_i\mathbf{D}_i\mathbf{w}_i)^\top\mathbf{I}$.

The multi-layer network is just a natural extension of the single layer network. We define the filter set $\mathbf{W}_k$ the filters of the k-th layer, and the corresponding receptive field matrix and lateral inhibition matrix are $\mathbf{D}_k$ and $\mathbf{L}_k$, respectively. The total filtering matrix in one layer, without considering non-linear mapping function, is defined by $\mathbf{F}_k = \mathbf{W}_k\mathbf{D}_k\mathbf{L}_k$. Then the response of the multi-layer network is defined by:

$$\mathbf{R} = \left(\prod_{i=1}^{k}\mathbf{F}_i\right)^\top \mathbf{I}$$

Since $\mathbf{F}$ is a linear transformation, we call $\mathbf{C} = \prod_{i=1}^{k}\mathbf{F}_i$ combined filter matrix. Each column of $\mathbf{C}$ is corresponding to one combined filter. The size of the receptive field of combined filters is increasing with the order of the layers. Thus, SHM gives a presentation of different scales in different layers. Furthermore, the staggered receptive fields give the representation in different positions over the input image. The presentations of different scales and different positions is the key idea of why SHM is better than other flat representations.

The structure of the proposed SHM is shown in Fig. 2.5. We only draw the weight connection that is non-zero.

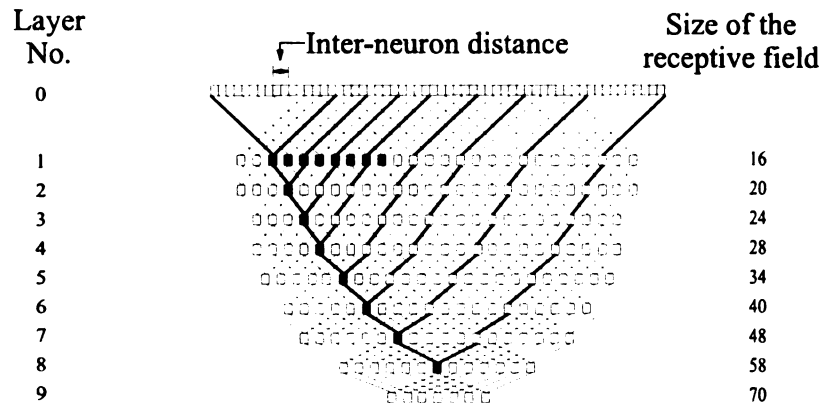


Figure 2.5: The architecture of SHM. Each square denotes a filter. Layer 0 is the input image. The filters marked in black in layer 1 belong to different eigen-group, because they overlaps. Bold lines that are derived from a single filter and expanded to the original image mark the receptive field of the combined filter. The size of the receptive field in a particular layer is 20% larger than its previous layer in this diagram, which is shown at the right. The size of the receptive field is rounded to the nearest integer.

2.3 Developmental Algorithm

The developmental algorithm for the SHM must be incremental and fast for real-time application. By "incremental" we mean that each input image must be discarded after it has been used for updating the SHM. It is not practical to store all of the images for batch processing due to the large amount of space required. By "fast" we mean that updating for each image must be computationally efficient. For example, iteration within each updating is not allowed. These requirements limit the complexity of the computation that can be performed by a biological cortex. They also make the design of a developmental program for SHM challenging.

We have further developed the CCIPCA (Weng et al., 2003) that was designed with these requirements as design constrains. This algorithm can be found in the appendix.

The filter development procedure of a single layer is described as follows:

Algorithm 1 The filter development algorithm of a single layer.

1: Use steps 1 to 2 of the CCIPCA algorithm in the appendix to initialize each filter in the network.
2: Initialize the residue image $\mathbf{I}_0 = \mathbf{I}(n)$, where $\mathbf{I}(n)$ is the n-th input image.
3: $k \leftarrow 0$;
4: **for all** filters $\mathbf{w}_i$ in filter group $\mathcal{G}_k$ **do**
5: Use Eq. A.2 of the CCIPCA algorithm to update connection weight vector $w_i(n)$ of each corresponding filter with residue image $\mathbf{I}_k$.

$$\mathbf{w}_i = \alpha(n)\mathbf{w}_i + \beta(n)(\mathbf{D}_i\mathbf{w}_i)^\top \mathbf{I}_k\mathbf{I}_k,$$

where $\alpha(n)$ and $\beta(n)$ are the amnesic average coefficient defined in Eq. A.3.
6: Use Eq. A.4 to compute the corresponding response $r_i(n)$ of each filter,

$$r_i = \frac{(\mathbf{D}_i\mathbf{w}_i)^\top \mathbf{I}}{\|\mathbf{D}_i\mathbf{w}_i\|}$$

7: Use Eq. A.6 to compute the residual image for the next eigen-group,

$$\mathbf{I}_{k+1} = \mathbf{I}_k - \sum_{\mathbf{D}_i \in \mathcal{G}_k} \frac{(\mathbf{D}_i\mathbf{w}_i)^\top \mathbf{I}_k (\mathbf{D}_i\mathbf{w}_i)}{\|\mathbf{D}_i\mathbf{w}_i\|^2}.$$

8: **end for**
9: $k \leftarrow k + 1$; Go to step 3 until all the eigen-groups are updated.
10: Get the next input image $I(n + 1)$ and go to step 2.

Each layer is similar in that it takes its input and runs the CCIPCA algorithm on it, producing both filters for this layer and the output response image for the next layer. The first layer of the network uses the original input image, and each subsequent layer uses output response images from the previous layer.

The receptive field in the same level is not concentrated in only several center positions. Instead, we use a masking matrix $\mathbf{D}_i$ to spread the receptive fields all over the input layer. Then the staggered positions, along with the increasing size of the receptive field, make it possible for input regions to have local representations with any size at any position (up to a sampled resolution) in the network.

Inhibitions between neurons play a vital role in the function of the visual cortex. We use residual images to represent the effect of inhibitive competition between neurons. This SHM network is not a feed-forward network because it deals with the re-entrant process of neurons. Each time a filter is computed, it is subtracted out from the original image, leaving information that has not yet been extracted by the current eigen-group. The next filter can only "see" part of the input that the previous filters cannot extract, thus, the first one can inhibit the latter one, which makes the process asymmetric. Each filter can "see" unique input data that has not been "seen" by previous neurons. This is more efficient than symmetrical inhibition where it takes time to compete the order and achieve the same result.

In SHM, the lateral interaction is not explicitly modeled by an lateral interaction matrix $\mathbf{L}$. Instead, filters are lateral affected by the residue images process. Suppose we represent $\mathbf{F}_i = \mathbf{D}_i\mathbf{w}_i$, then the response of the filter r_m, where $\mathbf{D}_m \in \mathcal{G}_k$, can be expressed as,

$$r_m = (\mathbf{F}_m)^\top \left[\mathbf{E} - \sum_{F_i \in \mathcal{G}_{k-1}} \mathbf{F}_i \mathbf{F}_i^\top \left(\mathbf{E} - \ldots \left(\mathbf{E} - \sum_{F_i \in \mathcal{G}_1} \mathbf{F}_i \mathbf{F}_i^\top \right) \ldots \right) \right] \mathbf{I}$$

The term in the square brace is the transpose of the lateral interaction matrix $\mathbf{L}^\top$.

Fig. 2.6 shows a sequence of residual images. It can be seen that the variance of the series residual images is decreased sequentially as more orthogonal components are extracted.

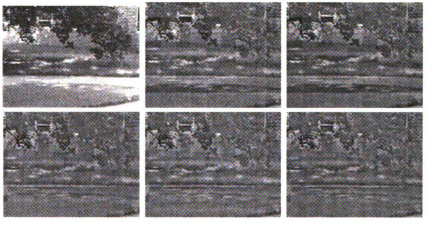


Figure 2.6: Sequence of residual images. From left to right and top to bottom starting from the original input image, the sequence of images displays the process that a component is subtracted from the previous one.

2.4 Computational complexity analysis

We would like to give a concise expression for the amount of computation. Since it is unnecessary to keep the same inter-filter distance for all areas, we can assume a similar number of connections in every cell in the sensory mapping. The number of

processing elements is roughly the same across different layers, which can be estimated from input dimension d.

Suppose that in layer A_i the average number of neural connections per filter is c_i and the array of filters has h_i rows and w_i column. The amount of computation required to update connections using incremental Hebbian-like algorithm is linear in c_i. Thus, with L layers, $i = 1, 2, ...L$, the amount of computation in the entire network is:

$$T_s = \sum_{i=1}^{L} h_i w_i O(c_i) \approx O(dcL),$$

where d is the number of pixels in the sensory input, c is the average number of connections per filter.

Since the sensory mapping stores the information in the place where cell computation and processing result needs to be stored too, the space complexity is equal to:

$$S_s = O(dcL).$$

2.5 Input Image Reconstruction

With SHM the input image is coded as responses at several layers. To show how complete the representation is at each single layer, we need to reconstruct the original image from the response of that layer alone. The reconstruction is a reverse processing of the mapping. The responses of a certain layer, which are actually the sample's

projection along the principal components, are used to recover its component in the sample space, then added back to the residual image. The reconstructed image $\mathbf{I}_r$ is computed by,

$$\mathbf{I}_r = \sum_k \sum_{\mathbf{D}_i \in \mathcal{G}_k} r_i \left(\mathbf{D}_i \mathbf{w}_i\right), \tag{2.5}$$

where r_i is the response of the i-th filter.

Lemma 2.5.1 *The filters with localized receptive field in the same eigen-group, are orthogonal to each other.* $\forall$ *filters* $\mathbf{D}_i, \mathbf{D}_j \in \mathcal{G}$, $i \neq j$ *we have* $< \mathbf{D}_i \mathbf{w}_i, \mathbf{D}_j \mathbf{w}_j >= 0$

The notation$< \cdot >$ is defined as inner product.

Proof. $< \mathbf{D}_i \mathbf{w}_i, \mathbf{D}_j \mathbf{w}_j >= \mathbf{w}_i^\top \mathbf{D}_i^\top \mathbf{D}_j \mathbf{w}_j$. From Eq. 2.2, we have $\mathbf{D}_i^\top \mathbf{D}_j = \Theta$, where Θ is zero matrix. Therefore, $< \mathbf{D}_i \mathbf{w}_i, \mathbf{D}_j \mathbf{w}_j >= 0$. $\triangleleft$

Lemma 2.5.2 *The residue vector* $\mathbf{I}_k$ *of an Image* $\mathbf{I}_{k-1}$ *after applying eigen-group* $\mathcal{G}_k$ *is orthogonal to every filter in the* $\mathcal{G}_k$, *i.e.* $\forall \mathbf{D}_i \in \mathcal{G}_k, < \mathbf{I}_k, \mathbf{D}_j \mathbf{w}_j >= 0$, *where* $\mathbf{D}_j \in \mathcal{G}_k$.

Proof. From Eq. 2.4, it is simple to show

$$\begin{aligned}
< \mathbf{I}_k, \mathbf{D}_j \mathbf{w}_j > &= \left[\mathbf{I}_{k-1} - \sum_{\mathbf{D}_i \in \mathcal{G}_k} \left(\mathbf{D}_i \mathbf{w}_i\right)^\top \mathbf{I}_{k-1} \left(\mathbf{D}_i \mathbf{w}_i\right) \right]^\top \mathbf{D}_j \mathbf{w}_j \\
&= \left(\mathbf{D}_j \mathbf{w}_j\right)^\top \mathbf{I}_{k-1} - \sum_{\mathbf{D}_i \in \mathcal{G}_k} \left(\mathbf{D}_i \mathbf{w}_i\right)^\top \mathbf{I}_{k-1} \left(\mathbf{w}_i^\top \mathbf{D}_i^\top \mathbf{D}_j \mathbf{w}_j\right) \\
&= \left(\mathbf{D}_j \mathbf{w}_j\right)^\top \mathbf{I}_{k-1} - \left(\mathbf{D}_j \mathbf{w}_j\right)^\top \mathbf{I}_{k-1} = 0
\end{aligned}$$

Since $\mathbf{D}_j \in \mathcal{G}_k$, according to Eq. 2.2 the addend in the summation is zero when $i \neq j$, and $\left(\mathbf{D}_j \mathbf{w}_j\right)^\top \mathbf{I}_{k-1}$ when $i = j$. Note here we assume $\|\mathbf{D}_i \mathbf{w}_i\| = 1$.

◁

Therefore, the initial image space is partitioned into a sequence of subspaces, which is spanned by the vectors in each eigen-group. And each of these subspaces is orthogonal to the next residue image. It is then easy to show that the total reconstruction error is

$$\begin{aligned} E &= \|\mathbf{I} - \mathbf{I}_r\|^2 \\ &= \left\| \mathbf{I}_0 - \sum_{j=1}^{k} \sum_{\mathbf{D}_i \in \mathcal{G}_j} r_i \left(\mathbf{D}_i \mathbf{w}_i\right) \right\|^2 \\ &= \left\| \mathbf{I}_1 - \sum_{j=2}^{k} \sum_{\mathbf{D}_i \in \mathcal{G}_j} r_i \left(\mathbf{D}_i \mathbf{w}_i\right) \right\|^2 \\ &= \|\mathbf{I}_{k+1}\|^2 \end{aligned}$$

where $\|\cdot\|$ is the l_2 norm of the vector, and $\mathbf{I}_{k+1}$ is the last residue image after being processed by k eigen-groups.

We need also to consider the border conditions. If the localized filter covers the border of the input image, then the connections that are out of the image will get zero input. This condition will increase the image by $(z-1) \times 2$ rows and $(z-1) \times 2$ columns, as shown in Fig. 2.7. This new image space will have dimension

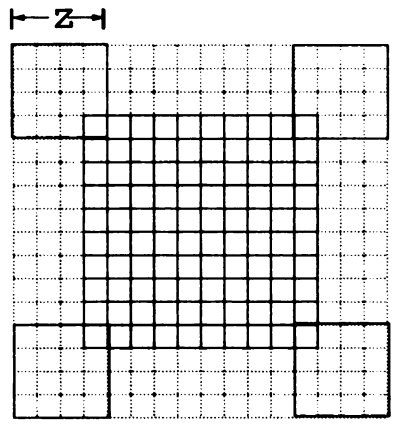


Figure 2.7: The filter connections that covers the outliers of the image with have zero input.

of $(r+2(z-1))(c+2(z-1))$. Since the outlier pixels are set to zero, the true dimension of the image space is still $r \times c$.

The total number of filters that have localized receptive field covering at least part of the input image will be $(r+2(z-1))(c+2(z-1))$. For all $\mathbf{w}_{pr}$ and $\mathbf{w}_{qs}$, we have $< \mathbf{w}_{pr}, \mathbf{w}_{qs} >= 0$. And the number of filters is more than the effective dimension of the image space. Thus, the filters are linearly dependent if the image border is also covered by filters.

The single layer reconstruction algorithm is shown in Algorithm 2.

Algorithm 2 The single layer reconstruction algorithm

1: $\mathbf{I}_r = \Theta$, where Θ is the zero matrix.
2: **for all** Response output r_i and the corresponding filter vector $\mathbf{w}_i$ and receptive field $\mathbf{D}_i$. **do**
3: Update the reconstructed image by,

$$\mathbf{I}_r = \mathbf{I}_r + \sum_k \sum_{\mathbf{D}_i \in \mathcal{G}_k} r_i \left(\mathbf{D}_i \mathbf{w}_i\right)$$

4: **end for**

2.6 Attention Effectors

The response of each filter (or processing cell) at every level is available to be used for later processing. The goal of the attention effectors is to suppress the response (value) from areas that are beyond the attended visual field.

However, the filters share input planes with the staggered receptive fields shown in Fig. 2.5. This means that any region of filters in a layer of SHM does not correspond to a clear-cut region in the input plane. A special case of this situation happens when this region is so small that it contains only one filter. Interactions between filters expand the extent of each pixel beyond that marked by direct inputs to filters. Therefore, we should not expect that selecting the response from a region in any layer will result in a clear-cut attended region in the input.

We define an attended region as a 3-D ellipsoid centered at a position (x,y,l) in a layer of SHM, where, x, y, denotes the position of the filter and l denotes the layer. The length of each axis of the ellipsoid corresponds to the scale along the dimension. All the filters falling into the ellipsoid have their output passed on and all others are blocked.

2.7 Experiment with SHM

Testing began with a collection of over 5000 natural images. They were taken using a Sony digital camcorder in an outdoor environment. A variety of recordings were taken, including buildings, trees and shrubbery, closeups of bark and wood chips, flowers, and other various scenes found in nature and on campus. The frames from

the video were then captured and digitized into 118×158 images (each has 118×158 pixels). Fig. 2.8 shows a few sample images from the image set.

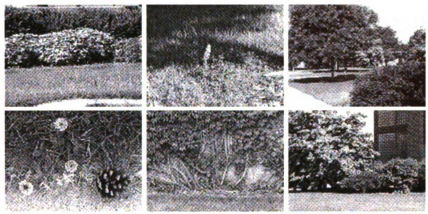


Figure 2.8: Several samples of 5000 natural images collected.

There are two methods used to develop filters, weight-sharing or non-weight-sharing. The weight-sharing method assumes that the filters that are in the same eigen-group have the same connection weights. This is a good estimation only if the images have the same statistical properties across the entire image frame. Thus, the sharing method only updates a single filter for one eigen-groups. The non-sharing method updates a separate set of filters for each filter in the eigen-group.

The filters of the first layer, trained by the sharing method, is shown in Fig. 2.9. There are 64 eigen-groups in this layer and the size of each filter in this layer is 16×16. The filters are reordered according to their computation order so that the dominant principal components are shown first. The first several filters appear similar to the receptive field excitation-inhibition patterns reported by Hubel and Wiesel's experiment (Hubel and Wiesel, 1962). Later ones show a complex pattern, which may fall into the cell classes that were not categorized by Hubel and Wiesel.

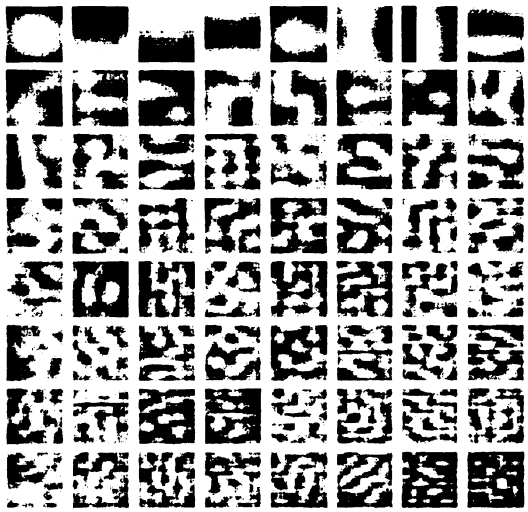


Figure 2.9: The filters developed in the first layer by the sharing method. The order of dominance is from left to right and from top to bottom.

When input was placed into the network, a scalar output response value was generated for each filter at its respective receptive field. This measures the amount of energy of the input along the direction of the filter and represents the firing strength of a filter. The filter responses were then organized into groupings based on their order in the eigen-group and combined into a structured image vector and passed on as input to the next layer of the network.

The responses of a layer are reconstructed by its following layer, finally reaching the original input layer. The reconstructed images are shown in Fig. 2.10, which indicated that the response of each single layer is sufficient to reconstruct the original image fairly well. We also test the reconstruction error of each layer. The chart in Fig. 2.10 shows the error. The error is computed based on the Euclidean distance between the original image and the reconstructed image. The two images are normalized by subtracting the mean and being divided by their standard deviation. The distance is then divided by the number of pixels to make it possible to compare im-

ages of difference sizes. The error is averaged over 100 reconstruction results. Fig. 2.11 illustrates the reconstruction error of each layer.

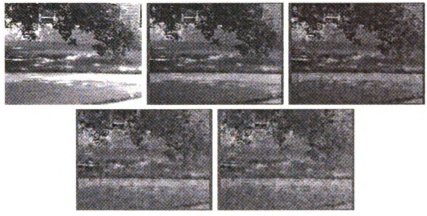


Figure 2.10: The reconstructed images from different levels. The first image is the original input image. Others are reconstructed images from Layer 1 to Layer 4 respectively.

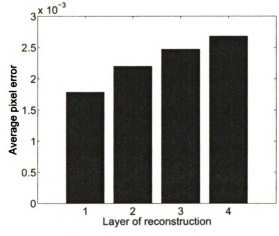


Figure 2.11: The reconstruction error of each layer.

2.8 Experiment with Occlusion

The purpose of the experiment summarized here is not to show a practical application system, but rather to study the use of SHM as a local representation suitable for

attention selection. The proposed sensory mapping provides a complete family of local representation that is needed for the generation of attention control signals from later processing. Further, the proposed sensory mapping serves as an effector of the top-down attention control. In a future system that uses SHM as a sensory "cortex," the parallel outputs of all filters in all layers are used by a higher level processing that generates attention signals for the SHM. There have been numerous studies on attention selection. However, due to the complex nature of attention selection, those studies use hand-defined task-specific saliency as the attention selection criteria. The architecture proposed here is a learning-based general mechanism at the sensory mapping level, for autonomously developed attention selection behaviors, potentially for any task. In the current experiment, we used a regressor denoted as $C3$,shown in Fig. 2.12, that learns the mapping from input to the attention control signals needed by the SHM at any given time. This regressor is trained first.

The experimental setting is organized as follows: In the training phase, a series of global views of face images is presented to the system with class labels. In the testing phase different face images of the same set of people are presented to the system, but all of them are partially occluded: the upper view (U view) of a face has the lower third area replaced by a uniform intensity (gray), and the lower view (L view) has the upper third replaced by gray.

If a system is passive, it learns the global view and tries to match the input U or L view with the learned global prototype. If we use the nearest neighbor method (NN) to find the prototype, we call it monolithic + NN testing.

We also study the use of active internal action in visual perception. For this

purpose, we construct an active system shown in Fig. 2.12. Three classifiers $C1$, $C2$ and $C3$ are integrated in the system. Here, we use the Hierarchical Discriminant Regression Tree (HDR Tree) (Hwang and Weng, 2000). To learn the mapping, $C3$ is to determine what specific occlusion case the image is, U or L. In our test, the classifier $C3$ has reached a 100% classification rate.

The $S1$ and $S2$ are SHMs. $S1$ is for U views and $S2$ is for L views. $C1$ and $C2$ are two HDR Trees used to classify images. Which SHM + HDR should be used is determined by the attention signal from $C3$.

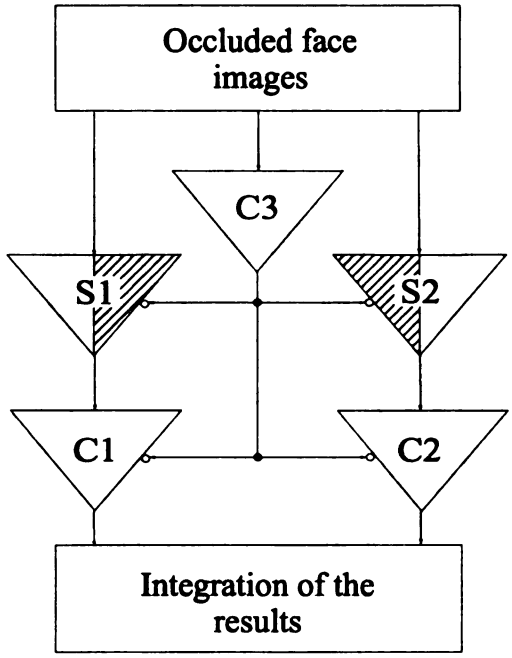


Figure 2.12: The schematic illustration of operation. C3 is the attention signal generator. C1 and C2 are the classifiers for each occlusion case. S1 and S2 are SHMs, S1 for U views and S2 for L views.

The experiment used the face data from the Weizmann data set. The set was taken from 28 human subjects, each having 30 images with all possible combinations of two different expressions, three lighting conditions and five different facial orientations.

Method	Recognition Rate		
	U	L	U+L
Monolithic+NN	51.43%	75.83%	82.38%
SHM+HDR	92.86%	95.95%	98.57%
Method	Testing Time (ms)		
	U	L	U+L
Monolithic+NN	765.5	765.5	2263.0
SHM+HDR	702.4	702.4	2016.6

Table 2.1: Summaries of the occlusion experiment.

We use the leave-one-out cross-validation method. 30 experiments were conducted, in which 1 out of 30 samples of each person is picked up as a testing sample, and all of the remaining ones as training samples. The average of the recognition result is reported in Table 2.1, along with the time run on a dual Pentium III 600 MHz processor PC. The U, L columns are for testing with U, L views only. The U+L column is for an integration of 2 test views, U and L are to give a single classification (person). The U+L integration is done in the following way. For each view, top K best matches form a top labels list. Thus, each pair of U and L views give a U label list and a L label list. The label that has the minimum rank sum is the output label.

Table 2.1 shows that the proposed SHM with the HDR classifier is effective in integrating active partial views, U and L. Without active attention, monolithic + NN performed poorly, even with U+L integration. This is because it lacks a mechanism to actively extract local views when presented with a global view. The speed of the SHM + HDR is fast enough to reach about 2 Hz refreshing rate.

2.9 Experiment with Navigation Simulation

We also applied the SHM algorithm to a navigation simulation task. A total of 18,330 images are collected from two cameras mounted on an autonomous vehicle. The images of the camera are aligned and merged together to offer a broad view of the environment. A sample image with dimension 120 by 320 pixels is shown in Fig. 2.13.

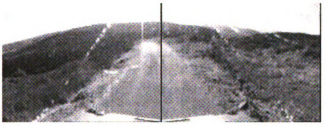


Figure 2.13: Merged image sample from the two cameras.

2,541 images from the data set are used to train a three level SHM network. The filter size of each layer is 16, 8, 8, starting from the bottom layer. Unlike the previous occluded face recognition experiment, a more general attention selection effector is used. As shown in Fig. 2.14, we use a GUI program to manually select 6 points on the road boundary, which are marked as circles. These 6 points are attention positions. A small window around the attention position and across levels of SHM is extracted. Let us denote $(S_1, S_2, S_3, S_4, S_5, S_6)$ as the concatenation vector of all 6 windows. The coordinates of the attention positions $(r_1, c_1, r_2, c_2, ..., r_6, c_6)$ are also concatenated with the input vector. Since these are different data modality from the SHM output, we have normalized each section with their respective variances. Thus,

the input vector to the HDR mapping is

$$(\frac{S_1, S_2, S_3, S_4, S_5, S_6}{\sigma_1}, \frac{r_1, c_1, r_2, c_2, ..., r_6, c_6}{\sigma_2}),$$

where σ_1 is the variance of the SHM output, and σ_2 is the variance of the attention point coordinates. This manipulation will make sure that the coordinate information will not be overwhelmed by the high dimensional image input. Three driving directions: going left, going right and going forward, are also provided by the GUI program as classification labels for the supervised learning. Note that the attention signals are all manually provided at this stage. Automatical attention selection will be the further research topic of the SHM model.

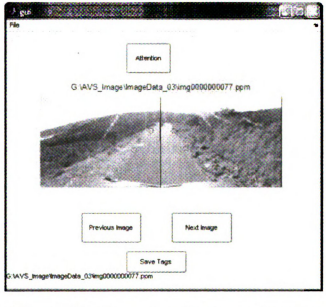


Figure 2.14: Six attention positions on the image.

We compare the results with the monolithic representation, in which only a flat image is used as input to the HDR mapping, and class labels remain the same. Since

Method	Recognition Rate
Monolithic+HDR	44.72%
SHM+HDR	85.52%
Method	Testing Time (ms) per image
Monolithic+HDR	103.2
SHM+HDR	748.1

Table 2.2: Summaries of the navigation image simulation.

the input image is of high dimension, we scale it down to one sixteenth of the original size. 2,541 images are used as training samples, and the rest 15,789 images are used as testing set. The result is compared in Table. 2.2. It clearly shows that the SHM representation is much better than the monolithic representation.

2.10 Conclusions

This chapter presents the design principles, architecture and developmental algorithm of a general purpose sensory mapping called the Staggered Hierarchical Mapping that tightly couples attention guided bottom-up information generation and execution of top-down attention selection. The experimental data has shown that the representation at each layer has a high completeness. The test results of the occluded image recognition system show that the SHM has a local analysis property, thus, it can be used to achieve attention control in an active vision system. A future research direction is to study mechanisms other than PCA develop sensory maps, as well as their advantages and disadvantages.

Chapter 3

Lobe Component Analysis

Orientation selective cells in V1 are well known, but the underlying computational principles that guide their emergence (i.e., development) are still illusive. The result reported here is based on high-dimensional probability distribution using a network structure. We introduce a concept called *lobe component.* Each lobe component represents a high concentration of probability density of the high-dimensional sensory inputs. A developmental algorithm for sensory network has been designed and tested based on the concept of lobe components. Our simulation result of network development revealed that many *lobe component cells* developed by the algorithm from natural images are orientational selective cells similar to those found in V1. Therefore, we make a hypothesis that orientation selection is only a natural consequence of cortical development but probability density estimation by neurons is a fundamental principle of cortical development. If this hypothesis is true, the principle of neuronal probability approximation may have deep implications to the development and self-organization of other sensory cortices (e.g., auditory and somatosensory cortices) and

higher cortices.

The proposed developmental algorithm is not meant to be biologically verified but is biological plausible. It is based on two well-known computational neural mechanisms, Hebbian learning and lateral inhibition. It is an in-place developmental algorithm in which every cell is both a signal processor and the developer of the cortex. There is no need for a separate network to deal with the development (and learning). The algorithm is purely incremental in the sense that there is no need for extra storage for information other than that can be stored and incrementally computed by the processing network itself.

To clarify technically, four types of algorithms have been considered with progressively more restrictive conditions: batch (Type-1), block-incremental (Type-2), incremental (Type-3) and covariance-free incremental (Type-4). The proposed Candid Covariance-free Incremental (CCI) Lobe Component Analysis (LCA) algorithm seems the first Type-4 algorithm for independent component analysis (ICA). The preliminary simulation studies showed that it out-performed, in convergence rate and computation time, some well-known state-of-the-art ICA algorithms in Type-3 through Type 1 for high-dimensional data streams. The proposed LCA algorithm also has a simple structure, with the lowest possible space and time complexities.

In this chapter, we will first briefly introduce the background of Independent Component Analysis in section 3.1, which is closely related to the LCA method. Then, we proposed the method in section 3.2 and analyze its performance.

3.1 Introduction to Independent Component Analysis

Recently, there has been a resurgence of interest in independent component analysis (ICA). ICA has a wide application in signal and image processing, telecommunication, and medical data processing.

The basic assumption of ICA is that the signal we have observed from the sensory input is a *linear* combination of several latent independent sources [1]. The main purpose of ICA is to find latent independence from the signal as much as possible, without the priori knowledge of their distributions and the mixture model.

In contrast to Principal Component Analysis (PCA) method, ICA will consider the information that is beyond the second order statistics (covariances). In PCA, all necessary information is in the covariance matrix. The principal axis can be found by doing eigen-decomposition on the covariance matrix. In other words, PCA only considers the pair-wise correlation. Many ICA methods, on the other hand, need a preprocessing procedure, which will remove the covariance information. This preprocessing, often referred as "whitening" or "sphering", will diagonalize the covariance matrix and normalize each principal component to the unit length. After whitening, the covariance matrix will become a identity matrix, thus no second order statistics will remain. Therefore ICA deals with the higher order statistics, and attempts to find a linear transform that will make the resulting components as statistically

[1] Some methods also considered the non-linear mixing case, which is out of the scope of this chapter.

independent from each other as possible.

Several different research communities are interested in the ICA problem. In the statistics community, several researches have been proposed to find the "interesting" projection of a multi-variant random variables to show the structure of the data. This approach, called "projection pursuit" (Friedman and Tukey, 1974; Friedman, 1987; Huber, 1985), tries to find the projection direction, so that the projected signal distribution will be the least like a Gaussian distribution, which is very similar to certain ICA approaches that try to maximize the non-Gaussianality.

Blind source separation (BSS) is a class of problems in the signal processing community. It aims at recovering unobserved signals or "sources" from observed mixtures, exploiting only the assumption of mutual independence between the signals. So it is obvious that BSS and ICA address the similar problem. Practical BSS problems are usually solved by the ICA method.

Another aspect to describe this problem is the sparse coding mechanism. The sparse coding problem can be found in the neuroscience literature (Field, 1994; Olshaushen and Field, 1996). It aims at explaining the coding mechanism of the neural system. The sparse coding hypothesis suggests that the principal in the sensory coding is to produce a sparse-distributed representation of the sensory input. Sparseness here means the neuron will have a high probability to stay inactive, and also have a relatively higher probability for large response. Here, "relatively" means compared to a Gaussian distribution with same variance. While considering a family of neurons, this means that at each time instance only a small number of neurons will have a large firing rate, and the rest of them have low activity. Since sparseness is essentially

the property of a super-Gaussian distribution, maximizing the sparseness equals to maximizing the super-Gaussianality. Thus the sparseness principle finally leads to the solution of independent super-Gaussian components. In this sense, the sparse coding is merely a special case of ICA.

In this chapter, we will first define the ICA problem, then survey the related ICA works. A new ICA algorithm is proposed in this chapter. We also discuss the experiment result of the new method and its limitation.

3.1.1 Problem statement and assumptions

Cocktail party problem

To illustrate the problem of independent component analysis, let us start from a real world example. Suppose there are n people speaking simultaneously in a room. At the same time, n microphones are set up to pick up sound signals. This scenario is illustrated in Fig. 3.1. Obviously, the microphone input is a mixture of sound of all the people. Since the sound strength is inversely proportional to the square of the distance from the sound source, the mixing factor of each sound source is different. The problem is how to separate n original source signals from the n mixed signals. This problem is often called the "cocktail party problem."

The difficulty of this problem lies in that we don't know any priori knowledge, e.g. statistical property, of the speaking person, nor do we know the setting of the microphones, e.g. how far away each microphone is from each person. The only thing we know is that each sound source is independent from each other. Of course in

Figure 3.1: The cocktail party problem.

semantic level the conversations may not be independent: speakers may all talk about the weather. But in signal level, since every person has independent vocal organs, each sound source can be think as independent. Knowing independence does not help us much, because there is so much arbitrariness. It is also worth noting that this is only a simplified version of the actual cocktail party problem. We have assumed that there were no noises, no echo effects, and all sound signals reach microphones instantaneously.

To formalize the problem, let us denote the source signal by $s_1(t), s_2(t), ..., s_n(t)$, and the mixed signal by $x_1(t), x_2(t), ..., x_n(t)$. As shown in Fig. 3.1, the extenuating factors of the ith microphone to each of the jth sources are denoted by a_{ij}, which are unknown and inversely proportional to the square of the distances between source and sensor. Suppose microphones do not introduce any non-linearity. The mixed

signals $x_1(t), x_2(t), ..., x_n(t)$ can be expressed as:

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t) + ... + a_{1n}s_n(t)$$

...

$$x_n(t) = a_{n1}s_1(t) + a_{n2}s_2(t) + ... + a_{nn}s_n(t).$$

To simplify the notation, we can write above equations into a matrix form:

$$\mathbf{x}(\mathbf{t}) = \mathbf{A}\mathbf{s}(\mathbf{t}), \tag{3.1}$$

where $\mathbf{x}(\mathbf{t}) = [x_1(t), x_2(t), ..., x_n(t)]^\top$ is the mixed signal vector, $\mathbf{s}(\mathbf{t}) = [s_1(t), s_2(t), ..., s_n(t)]^\top$ is the source signal vector, and

$$\mathbf{A} = \begin{bmatrix} a_{11} \cdots a_{1n} \\ \vdots \quad \ddots \\ a_{n1} \cdots a_{nn} \end{bmatrix}$$

is the mixing matrix. The mixing is assumed to be instantaneous, which assumes the sources arrive at different sensors at the same time without delay. Since we have this assumption, the time index t can be dropped to simplify the notation. Then the mixture model becomes $\mathbf{x} = \mathbf{A}\mathbf{s}$.

Apparently if $\mathbf{A}$ is known, $\mathbf{s}$ can be easily solved by a set of linear equations. Even prior knowledge, e.g. probability density function (pdf), of source signal $\mathbf{s}$ can be useful for estimation the unknown parameters, e.g. to be used in a maximum likelihood estimation method. Unfortunately, $\mathbf{s}$ and $\mathbf{A}$ are both unmeasurable, which

makes this problem considerately more difficult. However, the recently developed method of Independent Component Analysis can be used to estimate **A** based on the information of independence. The goal of ICA is to find a linear transformation **W** for the dependent sensory signals **x** that makes the recovered signal **u** as independent as possible:

$$\mathbf{u} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{A}\mathbf{s}, \tag{3.2}$$

where **u** is an estimation of the sources. The sources **s** will be fully recovered when **W** is the inverse of **A** up to a permutation and scaling factor.

In the following sections we will review the general principles that lead to the solution of this problem.

What is independence

The definition of statistical independent is defined as follows (Tucker, 1962).

Definition 3.1.1 *Let C denote a collection of events. The events in C are said to be* independent *if the probability of the joint occurrence of any finite number of them equals the product of their probabilities.*

Basically, this is said that if a random vector **x** that consists of n random variables $x_1, x_2, ..., x_n$ with the joint pdf function $p(\mathbf{x})$ is equal to the multiplication of the marginal pdf's, i.e.:

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i),$$

then $x_1, x_2, ..., x_n$ are independent. A set of random variables is mutually independent if and only if the joint pdf $p(\mathbf{x})$ is *factorizable* in the above way.

A weak form of independent random variables is uncorrelated random variables. The uncorrelation is defined as:

Definition 3.1.2 *Two random variables x and y are said to be* uncorrelated *if their covariance is zero:*

$$\text{cov}(x, y) = 0.$$

An important property of independent random variables is that:

Theorem 3.1.1 *Independent random variables are uncorrelated.*

The proof of this theorem is relatively straight forward, and can be seen in much classical literature. For the sake of convenience, we have shown it here.

Proof. We first show this theorem holds for the case of two random variables. The extension to the n variable case is natural. We need to show that $\text{cov}(x, y) = 0$, where x and y are independent random variables. This is equivalent to showing $E(xy) = E(x)\,E(y)$. Since

$$\begin{aligned} E(xy) &= \int\int xyp(xy)\,dxdy = \int\int xyp(x)\,p(y)\,dxdy \\ &= \int xp(x)\,dx \cdot \int yp(y)\,dy = E(x)\,E(y), \end{aligned}$$

◁

It is worth noting that the inverse of theorem 3.1.1 is not true, which means the uncorrelated random variables are not necessarily independent, with the only

exception of Gaussian random variables. So the uncorrelation is only a necessary condition of independence. Many ICA algorithms have a preprocessing stage, which will enforce the uncorrelation of the data. Although this processing will not generate independent component, it greatly reduces the uncertainty in the ICA model and results in a fast convergence.

The independence and uncorrelation can be intuitively explained by visualization of the data, as we will show in section 3.1.1.

Super-Gaussian, sub-Gaussian and Kurtosis

Random variables can be classified based on their similarity to the Gaussian distributions, or Gaussianality.

There is a class of random distributions, whose center of probability density funtion is more peaked than a Gaussian distribution. And the probability density of its outliers is considerably higher than Gaussian pdf. This class of distributions is often called super-Gaussian distributions. Generally, the super-Gaussian distributions are also referred as "heavy tail" distributions, because of the higher density at the outliers.

Another class of random distribution, on the other hand, is more "shallow" at the center and has lower density at outliers than Gaussian distribution. Accordingly, they are referred to as sub-Gaussian distributions. Fig. 3.2 shows three typical random distributions, respectively the Laplace distribution, the Gaussian distribution, and the uniform distribution.

The Laplace distribution is also called the double exponential distribution. It

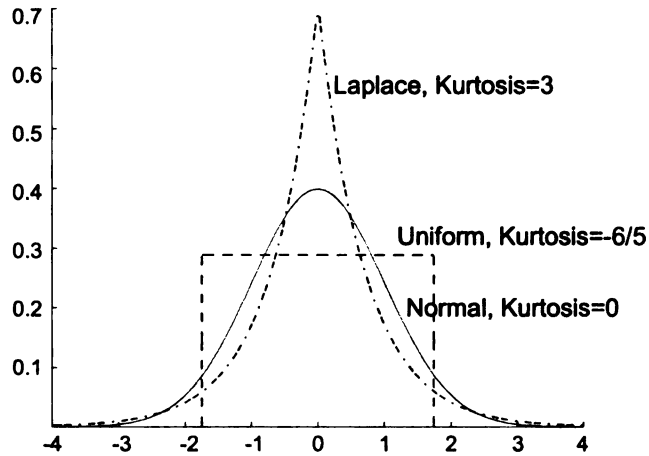


Figure 3.2: The density function of super-Gaussian, Gaussian, and sub-Gaussian distributions. All the distribution are normalized to unit variance. The Laplace distribution has positive kurtosis, thus is a super-Gaussian distribution. the uniform distribution has negative kurtosis, which implies sub-Gaussian. And the normal distribution has a zero kurtosis.

is the distribution of differences between two independent variables with identical exponential distributions. The pdf of the Laplace distribution is:

$$p(x) = \frac{1}{2b} e^{-|x-\mu|/b}, \tag{3.3}$$

where μ is the mean, and b controls the shape of the distribution. The Laplace distribution is a super-Gaussian distribution.

In Fig. 3.2, all three distributions are normalized so that their variance are all 1. This makes comparing kurtosis possible, which is a quantitative measurement of Gaussianality, since the kurtosis of distributions are comparable only when they have

the same variance. The classical kurtosis definition is:

$$kurt\,(y) = E\left\{y^4\right\} - 3\left(E\left\{y^2\right\}\right)^2, \tag{3.4}$$

where E denotes the expectation. If we assume y is of unit variance, the kurtosis is simplified to $E\left\{y^4\right\} - 3$. the kurtosis can be either positive or negative, which is actually a normalized forth order cumulant. The distributions that have positive kurtosis is called *super-Gaussian* distributions, and those with negative kurtosis are called *sub-Gaussian*. The Gaussian distribution and certain non-Gaussian distributions have zero kurtosis. In statistical literature, they are also referred as *leptokurtic*, *platykurtic*, and *mesokurtic* respectively.

Illustration of independence

Intuition is very important for understanding complex and abstract linear algebra problems. In this section, we try to explain the ICA problems with intuitive figures. We focus on examples in the 2-D data space, since they can be easily plotted.

Suppose we have a random vector $\mathbf{s} = [s_1, s_2]^\top$, where the random variables s_1 and s_2 are Laplace distribution, whose probability density is defined as in Eq. 3.3.

Fig. 3.3 shows the joint distribution of these random variables. The x and y coordinates of each data sample denote the values of s_1 and s_2. The dot-dash lines represent the source component axis.

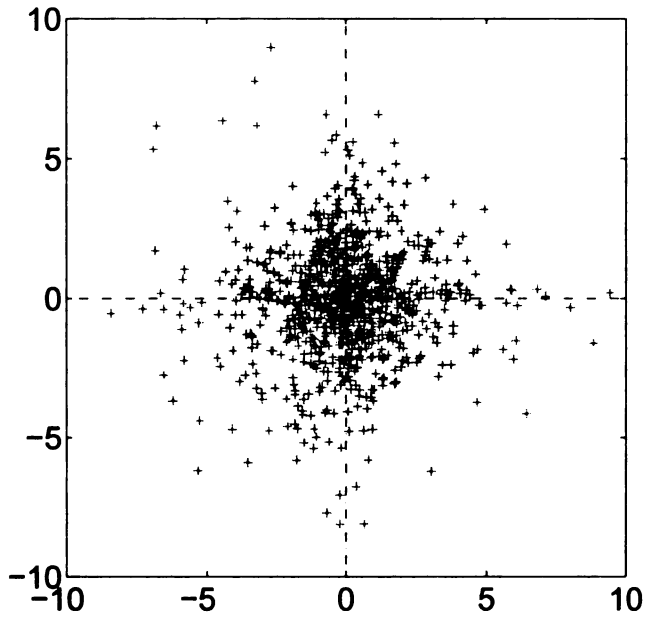


Figure 3.3: Two Laplace distribution.

The mixing matrix **A** is defined as:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}. \tag{3.5}$$

By applying the mixing matrix **A** to the source signal, we will get the mixed signal $\mathbf{x} = \mathbf{As}$. Fig. 3.4 displays the mixed distributions. Since the mixing matrix is a linear transformation matrix, it performs rotating and scaling transformation to the sample data points. The dot-dash lines in Fig. 3.4 are original independent components transformed by the same mixing matrix. Then the task is to find the independent components given those mixed data points.

Fig. 3.5 shows the example of sub-Gaussian distribution under the same mixing matrix as in Eq. 3.5. The source random distributions are uniform distribution of

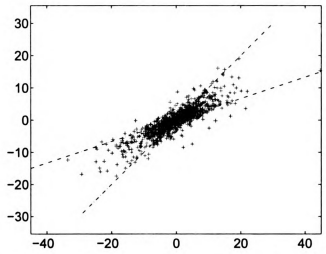


Figure 3.4: Two Laplace distribution mixed by mixing matrix **A**.

unit variance with the following pdf:

$$p(s_i) = \begin{cases} \frac{1}{2\sqrt{3}} & \text{if } |s_i| \leq \sqrt{3}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

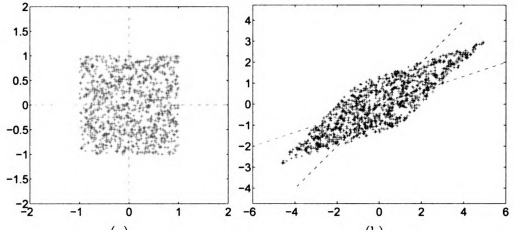


Figure 3.5: Example of two uniform distributions. (a) The source signal. (b) Two uniform distributions mixed by mixing matrix **A**.

The task of ICA is then to find the original source axis represented by the dot-dash

lines.

Preprocessing procedure

The central problem of ICA is to find the mixing matrix or its inverse form (demixing matrix). In the completed case (number of sources equals to the number of sensory inputs), the degree of freedom of the mixing matrix $\mathbf{A}$ is $n \times n$, where n is number of sources. This makes the searching for mixing matrix less tractable, since n^2 parameters need to be determined. However, by a carefully designed preprocessing procedure, the degree of freedom can be reduced, thus leads to not only a well conditioned problem but also a faster convergent speed for iterative approaches. In this section we discuss some common preprocessing strategies for ICA.

Centering: Subtracting the sample mean may be the commonest preprocessing. It is defined as: $\hat{\mathbf{x}} = \mathbf{x} - E\{\mathbf{x}\}$. This process will center the sample cluster and remove the possible shifting transformation from the mixing matrix $\mathbf{A}$. To fully recover the original source signal, the mean $E\{\mathbf{x}\}$ can be added back to the recovered signal $\mathbf{u}(\mathbf{t})$. In the following sections, if not specified we assume the sample vector $\mathbf{x}$ has already been centered.

Whitening: Whitening is an very useful preprocessing procedure. As we mentioned in the preceding section, the mixing matrix $\mathbf{A}$ performs both rotation and scaling transformation. In fact, the whitening procedure will remove the scaling transformation and only leave the unknown rotation transformation to the ICA step. So it will reduce the degrees of freedom of mixing matrix to $n(n-1)/2$ [2], about a

[2]The degrees of freedom of an $n \times n$orthogonal matrix is $n(n-1)/2$.

half of the n^2 degrees of freedom in an arbitrary mixing matrix.

Whitening is well established and can be achieved by eigenvalue decomposition (EVD) of the covariance matrix $E\left\{\mathbf{x}\mathbf{x}^\top\right\} = \mathbf{V}\mathbf{D}\mathbf{V}^\top$, where $\mathbf{x}$ is the sample vector, E denotes the expectation, $\mathbf{V}$ is the orthogonal matrix with columns as eigenvectors of the covariance matrix, and $\mathbf{D} = diag(d_1, ..., d_n)$ is the diagonal matrix of the eigenvalues. Whitening is achieved by:

$$\tilde{\mathbf{x}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^\top\mathbf{x}. \tag{3.7}$$

After whitening, the covariance matrix of $\tilde{\mathbf{x}}$ will be an identity matrix. Intuitively, whitening is equivalent to rotating the original axis to the principal component direction and scaling along the axis with the square root of the eigenvalue. Therefore the whitened vectors $\tilde{\mathbf{x}}$ have unit variances for every components $x_1, ...x_n$ and are decorrelated. Whitening is also referred as "sphering" due to this reason.

Fig. 3.6 shows the after-whitening effect of joint distribution of the mixed signal in Fig. 3.4 and Fig. 3.5(b). With an (up to now) unknown rotation transformation, we can then recover the original signal.

Why not Gaussian?

One of the basic assumptions of ICA is, to avoid ambiguity, at most one source could have a Gaussian distribution. This is because for Gaussian joint distributions decorrelation means independence. The whitened Gaussian distribution has central symmetry, so any set of orthogonal base could be the independent components. As

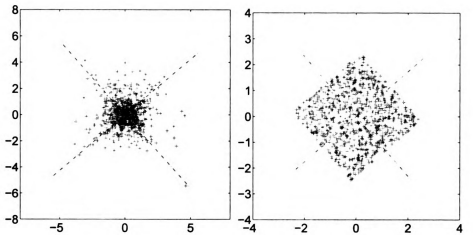


Figure 3.6: After whitening, the joint distribution is sphered. (a)Laplace joint distribution. (b)Uniform joint distribution.

shown in Fig. 3.7, the density is completely symmetric, then any two orthogonal axis could be the independent basis.

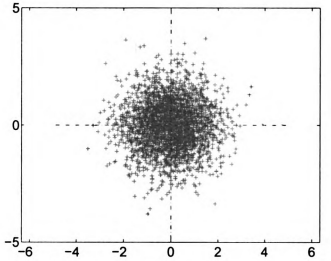


Figure 3.7: Joint distribution of two uni-variance Gaussian random variables.

More rigorously, we need to prove that upon arbitrary orthogonal transformation a Gaussian random vector with identity covariance matrix will yield same probability density.

Theorem 3.1.2 *Suppose* $\mathbf{y}$ *is a multi-variant random vector and* $\mathbf{y} = \mathbf{Mx}$, *where* $\mathbf{M}$ *is an* $n \times n$ *orthogonal matrix and* $\mathbf{x}$ *is a Gaussian random vector with zero mean identity covariance matrix, then* $\mathbf{y}$ *have the same probability density function as* $\mathbf{x}$.

Proof. In general, for any random vector $\mathbf{x}$ with density p_x and for any matrix $\mathbf{M}$, the density of $\mathbf{y} = \mathbf{Mx}$ is given by $p_x\left(\mathbf{M^{-1}y}\right)\left|\det \mathbf{M^{-1}}\right|$ (Papoulis, 1991). Since $\mathbf{x}$ has zero mean and identity covariance, its pdf is

$$p_x(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}} \exp\left[-\frac{1}{2}\mathbf{x}^\top \mathbf{x}\right], \tag{3.8}$$

then

$$\begin{aligned} p_y(\mathbf{y}) &= p_x\left(\mathbf{M}^{-1}\mathbf{y}\right)\left|\det \mathbf{M}^{-1}\right| = \frac{1}{(2\pi)^{d/2}} \exp\left[-\frac{1}{2}\left(\mathbf{M}^\top \mathbf{y}\right)^\top \left(\mathbf{M}^\top \mathbf{y}\right)\right] && (3.9) \\ &= \frac{1}{(2\pi)^{d/2}} \exp\left[-\frac{1}{2}\mathbf{y}^\top \mathbf{y}\right]. && (3.10) \end{aligned}$$

Then $\mathbf{y}$ and $\mathbf{x}$ has the same pdf function. ◁

3.1.2 Overview of ICA methods

The origin of ICA problem can be dated back to the 1980s, when Herault and Jutten (Herault and Jutten, 1986) proposed a feed forward neural network with feed back connections. The Herault-Jutten algorithm is based on a gradient descent updating rule that was able to separation independent sources, although the algorithm converges only under some severe restrictions, e.g. sub-Gaussian source signals. This work opened a remarkable chapter of independent component analysis and blind

source separation in the history and inspired many others.

The general framework of ICA introduced by Herault and Jutten was analyzed by Comon (Comon, 1994) in a more theoretical way. Comon proposed to minimize the mutual information of signals $\mathbf{u}$, where $\mathbf{u} = \mathbf{W}\mathbf{x}$ is an estimation of the source signals. In a different aspect, this is equivilant to minimizing the the Kullback-Leibler (Kullback and Leibler, 1951; Kullback, 1959) distance between the joint pdf and marginal pdfs. Since mutual information requires both the joint pdf of $\mathbf{u}$ and marginal pdf of $\mathbf{u}_i$, which are both unknown, one can only use some approximations to substitute the real probability density. Comon used a high order cumulant expansion of the Kullback-Leibler distance to approximate the probability densities.

In parallel to blind source separation research, unsupervised learning rules based on information theory are also considered by researchers in the neural network community. Linsker proposed a learning mechanism to maximize the mutual information between the input and output of a neural network (Linsker, 1992). Linsker did not explicitly connect it with ICA or blind source separation, but his method in some respects resembles the *infomax* principle independently proposed by other researchers (Bell and Sejnowski, 1995; Nadal and Parga, 1994), which is used to separate independent sources. The infomax approach is based on maximizing the information flow (mutual information between the inputs and outputs) of a neural network with non-linear output units. At about the same time, Maximum Likelihood Estimation (MLE) were proposed to recover the source independence (Pham et al., 1992). The MLE approach considers the mixed signals are generated by an statistical model with unknown parameter to be determined. Then the model parameters are estimated by

maximizing the likelihood of the observations. Strikingly, it is independently proved by several authors (Cardoso, 1997; Pearlmutter and Parra, 1997) that the MLE and infomax principle, regarding the source separation, are mathematically equivalent.

In the neuroscience community, the ICA problem is usually considered as a sparse coding problem. Sparseness is a characteristic of super-Gaussian random variables. Due to the central limit theorem, mixing several super-Gaussian random variables will decrease the sparseness of the mixed signals. So searching the independent direction to maximize the sparseness is a desirable strategy. Many studies have addressed the sparseness measurement issue and derived efficient ICA algorithms (Field, 1994; Olshaushen and Field, 1996; Hyvärinen, 2001).

In the following sections we will categorize the existing ICA methods and explain the details of each principles.

3.1.3 Methods based on Non-Gaussianality measurement

A straightforward way to recover independent components is measuring the similarity to the Gaussian distribution. The Central Limit Theorem tells us that the summation of n random variables is also a random variable, which has a Gaussian probability density when n approaches infinity. Intuitively, the summation of several non-Gaussian random variables are more "like" a Gaussian distribution than any of these individual random variables under certain conditions. So, maximizing the non-Gaussianality of the estimated signals $\mathbf{u} = \mathbf{W}\mathbf{x}$ with respect to $\mathbf{W}$ lead to the solution of the ICA problem.

Kurtosis

A quantitative non-Gaussianality measurement is kurtosis. The kurtosis of a random variable y is defined as $\text{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2$. Since in the ICA problem signals are generally pre-whitened and are of unit variance, the kurtosis turns out to be $\text{kurt}(y) = E\{y^4\} - 3$, which is a shifted fourth order cumulant. It is worth noting that the kurtosis is comparable only at the case of equal variance. For super-Gaussian random variables the kurtosis is generally greater than zero, while the sub-Gaussian is less than zero. The kurtosis is only defined on single variate random variable, therefore ICA based on kurtosis optimization is usually a single unit algorithm, which means every time the algorithm can only separate given a non-Gaussian signal.

The criteria function of this estimation is then defined as the kurtosis of estimated source signal $\mathbf{u}$, i.e. $J_{\mathbf{W}} = \text{kurt}(\mathbf{w}^\top \mathbf{x})$. The goal is to search the $\mathbf{W}$ space and to maximum or minimize the criteria function, in which the gradient ascent algorithm or other optimization approaches can be used. Hyvärinen and Oja has proposed a criteria function (Hyvärinen and Oja, 1997), which has the following form,

$$J(\mathbf{w}) = E\left\{\left(\mathbf{w}^\top \mathbf{x}\right)^4\right\} - 3\left\|\mathbf{w}\right\|^4 + F\left(\left\|\mathbf{w}\right\|^4\right), \tag{3.11}$$

where the first two terms on the right hand side represent the kurtosis. The last one is the penalty term to enforce the norm of $\mathbf{w}$ to one, since without this constrain the kurtosis can be arbitrarily large. The above criteria function can be optimized with standard gradient descent/ascent algorithm, but consequentially it will be affected by the selection of learning rate. In the same paper, Hyvärinen et.

al used a method related to Lagrange Multiplier and got an updating rule that only evolves computing the expectation, thus speeded up the convergence.

Kurtosis is a simple way to measure non-Gaussianality, however it suffers from several disadvantages. Kurtosis is a normalized fourth order cumulant, so it is very sensitive to the noise in the outliers. A few noise samples with high value will greatly change the kurtosis. Another drawback of kurtosis is that Gaussian distribution is not the only one that have zero kurtosis, so the kurtosis measurement can not recover the independent direction of these distributions.

Negentropy

An important conclusion in information theory is that the Gaussian random variables have the largest entropy of all possible random variables with the same variances. This indicates that the Gaussian random variables have the most randomness. In fact, the entropy is the most optimal criteria to measure non-Gaussianality.

The entropy for discrete random variable Y is defined as

$$H(Y) = -\sum_i P(Y = a_i) \log P(Y = a_i). \tag{3.12}$$

This definition can be generalized for continuous space random variables, in which case often called differential entropy, and defined as:

$$H(\mathbf{y}) = -\int f(\mathbf{y}) \log f(\mathbf{y}) dy. \tag{3.13}$$

Usually for convenience a modified version of differential entropy - negentropy is

often defined, which is

$$J(\mathbf{y}) = H(\mathbf{y_{Gauss}}) - H(\mathbf{y}), \tag{3.14}$$

where H denotes the differential entropy and $\mathbf{y_{Gauss}}$ is a Gaussian random variable. Negentropy is always non-negative, and is zero for Gaussian distributions. Negentropy is well justified by information theory and is also considered as an optimal estimation of non-Gaussianality. However, since it involves estimation of probability density, it is computationally expensive and practically intractable for the high dimensional random variables. This leaves the exact estimating of entropy a rather theoretical approach.

Many researchers considered the problem of entropy approximation (Comon, 1994; Hyvärinen, 1998) using high order cumulant. Comon used Edgeworth expansion based on third and forth order cumulant to approximate the negentropy as shown here:

$$J(\mathbf{y}) = \frac{1}{12}\kappa_3^2 + \frac{1}{48}\kappa_4^2 + \frac{7}{48}\kappa_3^4 - \frac{1}{8}\kappa_3^2\kappa_4 + o(m^{-2}), \tag{3.15}$$

where the $J(y)$ is the negentropy of random variable y and κ_n is the nth order cumulant. A simplified high order approximation can also be seen in the literature, which used only the first two term at the right hand side of Eq. 3.15 (Jones and Sibson, 1987). Using a high order cumulant to approximate negentropy encounters a similar problem as using kurtosis. The estimation is not stable and sensitive to the outlier noises.

To avoid these problems, Hyvärinen proposed using nonpolynomial cumulants to approximate the negentropy (Hyvärinen, 1998) based on maximum-entropy approach. The criteria function has the following form:

$$J(y) \approx \sum_{i=1}^{p} k_i \left[E\{G_i(y)\} - E\{G_i(v)\}\right]^2, \tag{3.16}$$

where the random variable v is a Gaussian random variable of zero mean and unit variance, and k_i are positive constants. It is worthy noting that this is also a generalized cumulant-based approximation. When we choose $G(x) = x^4$, Eq. 3.16 is degenerated to a normalized kurtosis.

Mutual information

Mutual information criteria is closely related to entropy. The mutual information I between m scalar random variables $u_i, i = ...m$ is defined as:

$$I(\mathbf{u}) = \sum_{i=1}^{m} H(u_i) - H(\mathbf{u}), \tag{3.17}$$

where $H(\mathbf{y})$ denotes differential entropy. Mutual information is always non-negative and equals to zero iff y_i are statistically independent. To find the ICA component, one needs to minimize the mutual information. A very important property of mutual information is if there exists an invertible linear transformation $\mathbf{W}$, so that $\mathbf{u} = \mathbf{W}\mathbf{x}$, then

$$I(\mathbf{u}) = \sum_{i=1}^{m} H(\mathbf{u}_i) - H(\mathbf{x}) - \log|\det \mathbf{W}|. \tag{3.18}$$

If we assume $\mathbf{x}$ is pre-whitened and $\mathbf{W}$ is only a rotation matrix, then the last two terms in right side of Eq. 3.18 will be independent from $\mathbf{W}$, thus the optimization of Eq. 3.18 is equal to minimizing the entropy, i.e. maximizing the negentropy. This shows the fundamental relation between the negentropy and mutual information.

Mutual information can also be explained by Kullback-Leibler divergence (Kullback and Leibler, 1951). The Kullback-Leibler divergence is defined as:

$$K(p\|q) = \int_{\mathbf{u}} \log\left(\frac{p(\mathbf{u})}{q(\mathbf{u})}\right) p(\mathbf{u}) d\mathbf{u}, \tag{3.19}$$

where p and q denote two probability densities. Intuitively, the K-L divergence can be thought of as the distance between two probability densities. Then if we substitute p with the joint probability density of $\mathbf{u}$ and marginal probability densities of $\mathbf{u_i}$, Eq. 3.20 is actually measuring the similarity between them:

$$\begin{aligned} K(f\,\|f_i) &= \int_{\mathbf{u}} \log\left(\frac{f(\mathbf{u})}{\prod_i f_i(\mathbf{u}_i)}\right) f(\mathbf{u}) d\mathbf{u} \\ &= \int_{\mathbf{u}} \log\left(f(\mathbf{u})\right) f(\mathbf{u}) d\mathbf{u} - \int_{\mathbf{u}} \log\left(\prod_i f_i(\mathbf{u}_i)\right) f(\mathbf{u}) d\mathbf{u}. \end{aligned}$$

by marginalize the second integrate in Eq. 3.19, we can see the K-L divergence is exactly same as mutual information.

3.1.4 High order cumulant based method

Statistical properties of the output data set u can be described by its moments or, more conveniently, by its cumulants $C^{(\mathbf{u})}$. Since the data have zero mean, the sample cumulants up to the fourth order are defined as

$$\begin{aligned}
\mathrm{C}_{\mathrm{i}}^{(\mathbf{u})} &= 0, \\
\mathrm{C}_{\mathrm{ij}}^{(\mathbf{u})} &= \left\langle u_i u_j \right\rangle, \\
\mathrm{C}_{\mathrm{ijk}}^{(\mathbf{u})} &= \left\langle u_i u_j u_k \right\rangle, \\
\mathrm{C}_{\mathrm{ijkl}}^{(\mathbf{u})} &= \left\langle u_i u_j u_k u_l \right\rangle - \left\langle u_i u_j \right\rangle \langle u_k u_l \rangle - \langle u_i u_k \rangle \left\langle u_j u_l \right\rangle - \langle u_i u_l \rangle \left\langle u_j u_k \right\rangle,
\end{aligned}$$

where $\langle \bullet \rangle$ denotes mathematical expectation. Cumulants of a given order form a tensor. The diagonal elements characterize the distribution of single components u_i. For example, the autocumulants of second, third, and forth order define *variance*, *skewness*, and *kurtosis*, respectively. The off-diagonal elements or cross-cumulants (all cumulants with $ijkl \neq iiii$) characterize the statistical dependencies between components. If and only if (iff) all components u_i are statistically independent, the off-diagonal elements vanish and the cumulant tensors of all orders are diagonal (assuming infinite amount of data) (Blaschke and Wiskott, 2002).

The ICA method based on high order cumulants is thus to find a linear transformation $\mathbf{W}$ that will minimize the absolute value of off-diagonal elements in the cumulant tensors of $\mathbf{u} = \mathbf{W}\mathbf{x}$ of all order. In practical terms, apparently infinite tensors are not tractable, therefore third and fourth order cumulants are usually used to approxi-

mate this problem (Comon, 1994; Blaschke and Wiskott, 2002). The advantage of this method is that it requires zero knowledge of the probability densities. Moreover, it is also used to approximate the mutual information as shown in section 3.1.3. Then it clearly shows the relation of high order cumulants and probability density. The order of cumulants used to approximate the problem is a trade-off between computational expense and accuracy.

3.1.5 Methods based on Maximum likelihood and Infomax

The goal of maximum likelihood estimation (MLE) is to model the observation $\mathbf{x}$ as generated from the latent random variables $\mathbf{s}$ via a linear mapping $\mathbf{A}$. In the noiseless case, we can use a parametric density estimator $\hat{f}(\mathbf{x};\theta)$ to find the parameter vector θ that maximizes the probability of observations $\mathbf{x}$. The estimator can be formulated as

$$L(\mathbf{W}) = \frac{1}{N} \log \prod_{t=1}^{N} \hat{f}_i \left(\mathbf{w}_i^\top \mathbf{x}(\mathbf{t}) \right) + N \log |\det \mathbf{W}| , \tag{3.20}$$

where $\mathbf{W}$ is the de-mixing matrix and is also treated as the parameter vector θ; N is number of observations; $\hat{f}_i$ is the marginal probability density of sources $\mathbf{u}$ with $\mathbf{W}$ as the parameter to be estimated (Pham et al., 1992; Pham and Garrat, 1997). Here we assume $\hat{f}_i$ is known up to a scaling factor here. The derivation of this MLE equation is due to the fact that: for any random vector $\mathbf{s}$ with density p_s and for any invertible matrix $\mathbf{W}$, the density of $\mathbf{x} = \mathbf{W}^{-1}\mathbf{s}$ is given by $p_s(\mathbf{W}\mathbf{x}) |\det \mathbf{W}|$ (Papoulis, 1991).

In parallel to the MLE study, unsupervised learning rules based on information theory were proposed by Linsker(Linsker, 1992). The basic idea here was to maximize

the mutual information between the output and signal portion in the input of a linear neural network. This principle was related to the redundance reduction principle suggested by Barlow (Barlow, 1961) as an coding strategy in neurons. By adding a non-linear transfer function to the network, Bell and Sejnowski (Bell and Sejnowski, 1995) proposed the infomax principle (information maximization), which used the stochastic gradient method to seek the maximization of mutual information between the input and output. A similar method was independently developed by Roth and Baram (Roth and Baram, 1996). Strikingly, the infomax and MLE are proved to be equivalent by Cardoso (Cardoso, 1997) and Pearlmutter and Parra(Pearlmutter and Parra, 1997) independently. They show that the MLE and the infomax can both be explained by the Kullback-leibler divervence between $\mathbf{Wx}$ and the estimated source $\mathbf{u}$.

The major problem of MLE and infomax principle is that they both assume that the marginal probability density of source $\mathbf{s}$ is known, which in most of the cases is unknown or intractable. In practise, both methods need an approximation of the density. For example, in Bell and Sejnowski's method they assumed a sigmoid function as the cumulative density function of the source random variables. Since the corresponding probability density function of the sigmoid function is a super-Gaussian distribution, Bell and Sejnowski's method can only deal with some super-Gaussian sources with questionable accuracy. Other researchers extended the infomax approach to address this problem.

Lee, Girolami and Sejnowski (Lee et al., 1999) suggested an extended infomax algorithm, in which they use mixture of Gaussian distributions to simulate the prob-

ability density of sub- and super-Gaussian signals and then are able to separate both type of signals.

The original infomax algorithm involves inverting the feedforward weight matrix $\mathbf{W}$, which makes the computation expensive. Amari et. al (Amari, 1998; Amari et al., 1996) and Cardoso et. al (Cardoso and Laheld, 1996) soon realized that it can be improved by using the natural gradient (or referred as relative gradient in Cardoso et. al's paper), which multiplies the gradient of the $\mathbf{W}$ matrix by a positive definite matrix $\mathbf{W}^\top\mathbf{W}$. This change will speed up the convergence by removing the matrix inversion.

3.1.6 Methods based on Sparseness measurement

As proposed by Olshausen and Field and others (Field, 1994; Olshaushen and Field, 1996; Olshaushen and Field, 1997), the cerebral cortex uses a sparse coding scheme. Statistically, given any input from the environment, only a few neurons respond actively. Most other neurons do not, or hardly, respond. The sparse coding is desirable since it effectively transforms an input event that is represented by many active pixels quickly down to a few neurons, which is similar to the redundance reduction principle suggested by Barlow (Barlow, 1961). The reason that sparseness is a reasonable conjecture, suggested by Olshaushen (Olshaushen and Field, 1997), lies in that natural scene can be usually described by a few structure primitives, such as edges, lines and curves. These features, if filtered with Garbor or wavelet filters, will show a sparse response pattern.

Olshausen and Field's suggested the following criteria function:

$$E(\mathbf{w}) = \sum_t \left[I(t) - \sum_i u_i(t)\mathbf{w}_i \right]^2 + \sum_i S\left(\frac{u_i(t)}{\sigma}\right), \tag{3.21}$$

where the first term at the right side of the equal sign is the reconstruction constrain, and second term is the sparseness measure. The reconstruction constrain is used merely for avoiding the trivial solution, which means all component vectors are zero vectors.

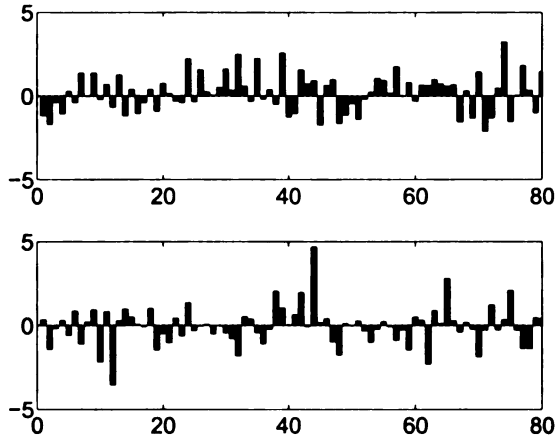


Figure 3.8: Gaussian signal vs. sparse signal. The sub plot on the top shows a zero mean unit variance Gaussian random variable. The X axis denotes different observations, and Y axis is the value. The bottom sub plot shows a sparse random variable with same mean and variance.

The sparse coding principle implies that the statistics of natural images is super Gaussian: it consists of a high probability region near the center, but with heavy tails. The high probability region near the center corresponds to most cases where inputs are nearly orthogonal to the feature that the neuron represents. The heavy tails correspond to the cases that the neuron represents and, thus, should respond.

How does a set of neurons detect super-Gaussianality? A common neural archi-

tecture is shown in Fig. 3.9.

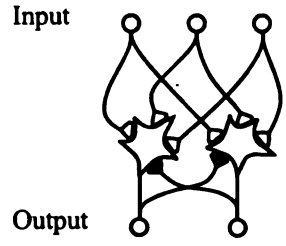


Figure 3.9: Two neurons with horizontal inhibitions. Hollow triangles indicate excitatory connections and solid triangles indicate inhibitory ones.

From the figure we can see that each neuron does not develop its weights alone. Consider a neuron, its response is inhibited by the response from nearby neurons that shares the similar receptive field. It also inhibits its neighboring neurons. If this neuron responses most strongly in the neighborhood, it most effectively inhibits its neighboring neurons, which further weakens the inhibition from its neighbors. After a few milliseconds, it effectively inhibits its neighbors, although not totally.

In section 3.2.8 we will propose an algorithm called Lobe Component Analysis (LCA) based on the sparseness measure. And in section 3.4, we will discuss the sparseness in more details.

3.1.7 Summary

ICA research has become very active. There has been at least one regularly scheduled conference series (Int'l Conf. on Independent Component Analysis and Blind Signal Separation) and several special issues (IEEE Proceedings, Oct. 1998; Signal Processing, vol. 73, 1999; IEICE Transactions, March 2003; Journal of Machine Learning Research, Dec., 2003; Neurocomputing, vol. 49, 2002, etc.). Many alternative methods

have been proposed, using the principles of minimum mutual information, maximum likelihood, infomax, higher-order cumulants such as Kurtosis, etc. (see a survey by Hyvarinen (Hyvärinen, 1999)).

In the following, we introduce four types of algorithm. **Type-1: Batch**. A *batch* algorithm collects observed data as a batch and then calculates the results. A batch algorithm cannot deal with very long or open-ended data streams and, therefore, cannot compute the results in real-time. **Type-2: Block-incremental**. A *block-incremental* algorithm is not allowed to store training data as a batch for computation. Instead, it can only store statistics, e.g., mean, covariance matrix and other higher order statistics for incremental computation. The block enables it to compute these statistics from the data in a temporal block (a small batch). A block-incremental algorithm can deal with open-ended data streams but the result is delayed by a block size. **Type-3: Incremental**. An *incremental* (also called sequential or adaptive) algorithm is a framewise incremental algorithm (block size is one time frame). By default, an *incremental* algorithm has a block size of 1 unless the term "block" is stated. An incremental algorithm must update its statistic estimates for one time frame at a time and, thus, it can process open-ended data streams with little delay. **Type-4: Covariance-free incremental**. Such an algorithm is an incremental algorithm, but further it is not allowed to compute the covariance matrix or other higher order statistics matrices. In summary, from Type 1 to Type 4, the conditions under which an algorithm is run becomes progressively more restrictive. In general, one should not expect that an algorithm of Type $(i+1)$ to converge faster than one of Type (i), $i = 1, 2, 3$.

The state-of-the-art batch algorithms include FastICA by Hyvarinen & Oja (Hyvärinen and Oja, 1997; Hyvärinen and Oja, 2000), which is among the fastest Type-1 ICA algorithms in terms of speed of convergence and its high capability to handle high-dimensional data. The Extended Infomax algorithm by Sejnowski and coworkers (Bell and Sejnowski, 1997; Lee et al., 1999) is a well-known Type-2 ICA algorithm. The NPCA-RLS algorithm by Karhunen (Karhunen and Pajunen, 1997) is a Type-3 ICA algorithm. We are not aware of any previously existing Type-4 ICA algorithm.

A Type-4 algorithm has several major advantages: It can process open-ended streams, causing little delay, and does not even store covariance matrix. The last property is especially important when the dimension d is large. Its potential relationship with the "in-place" development of a neuron will be discussed in Section 3.5 Broader Impact. By in-place development, we mean that an (artificial) neuron has to learn on its own while interacting with nearby neurons, to develop into a feature detector. A neuron with d connections does not have $d \times d$ storage space at its disposal. In other words, a Type-4 ICA algorithm for each ICA component is an in-place learner. We can expect that designing an effective Type-4 ICA algorithm is not trivial.

3.2 Lobe Component Analysis

Suppose a sequentially arriving series of vectors $\mathbf{x}_1, \mathbf{x}_2, ...$, where each input vector $\mathbf{x}_t$, drawn from a high-dimensional random space $\mathcal{X}$, is spatially highly correlated. It

is assumed that the temporal dependence of $\mathbf{x}_t$ is weak and not of interest. The basic need of our proposed research is to estimate the probability structure of $\mathcal{X}$.

Given an arbitrary high-dimensional space $\mathcal{X}$, the distribution of $\mathbf{x} \in \mathcal{X}$ may not necessarily have independent components. In other words, there exist no independent components $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_d$, so that their projection from $\mathbf{x} \in \mathcal{X} : \mathbf{x}_i = \mathbf{x}^\top \mathbf{w}_i$, $i = 1, 2, ..., d$, are statistically independent so that their p.d.f. is factorizable:

$$p(\mathbf{x}) = p_1(x_1)p_2(x_2)\cdots p_d(x_d).$$

Fig. 3.10 illustrates three cases. The first two cases are factorizable. The third case, a more general case, is not factorizable. In many high-dimensional applications (e.g., images and financial data streams), the p.d.f. is not factorizable.

3.2.1 Fine structure of high-dimensional density

High-dimensional density estimation is an important and yet very challenging problem which has been extensively studied in mathematical statistics, computer science and engineering. Major methods for probability density estimation include histograms, naive estimators, kernel estimators, nearest neighbor estimators, variable kernel methods, orthogonal series estimators, maximum penalized likelihood estimators and general weight function estimators (see, e.g., Silverman (Silverman, 1986) for a survey). For high-dimensional data, kernel-based methods, e.g., EM algorithms for a finite normal mixture (also called a mixture of Gaussians) (McLachlan, 1997), have been widely used.

These traditional methods are problematic when they are applied to practical high-dimensional data. The problems include: (1) The lack of a method for high-dimensional density estimation that satisfies the three stringent operative requirements: incremental, coveriance-free and undersample. By undersample, we mean that the incremental algorithm must work even when the number of samples is smaller than the dimension d. (2) The lack of an effective method to determine the model parameters (e.g., the means, covariances, and weights in the mixture-of-Gaussian models). (3) The lack of a method that gives a correct convergence (a good approximation for high-dimensional data), not just convergence to a local extremum (as with the EM method for mixture of Gaussians). (4) The lack of a method that is not only optimal in terms of the objective function defined, but also in terms of the convergence speed in the sense of statistical efficiency.

Approaches to estimating statistical distribution fall into two categories: global and local. *A global approach* uses shape characteristics of the distribution described by global higher order statistics. A global statistic is an expectation of the entire space of distribution, such as covariance matrix, kurtosis and other higher order cumulants (or moments).

A local approach estimates the distribution by decomposing the space of distribution into smaller regions, and each region is estimated by relatively lower order statistics (which is sufficient for local data). The kernel-based methods (e.g, the mixture-of-Gaussian methods) belong to the local approach. Techniques of global approaches can be used for local approaches.

A major advantage of a local method is to decompose a complex global modeling

problem into simpler local ones so that lower order statistics are sufficient. Higher order moments are sensitive to noise and outliers and are not effective for estimating a complex distribution. Local density approximation using lower-order statistics is more effective.[3] The challenges with a local approach include how to effectively decompose the global complex problem into local simpler ones and how to integrate solutions to local problems into the solution to the original global complex problem.

The proposed LCA uses a global-to-local (coarse-to-fine) estimation scheme. (1) First, we estimate the position: the mean $\bar{\mathbf{x}}$ of the data (through incremental estimation of $\bar{\mathbf{x}}$) and then subtract $\bar{\mathbf{x}}$ out of the observation. Thus, the transformed data has a zero mean. (2) Next, we estimate the scale: Incrementally estimate the covariance matrix of $\mathbf{x}$, and then whiten the zero mean observation so that it has a unit covariance matrix. (3) Estimate the lobe components from the zero-mean, unit covariance signals. Each lobe component represents a high concentration of p.d.f.. Without whitening the data, the drastic scale difference in different data components can obscure the fine details of the lobe components.

3.2.2 Global: Whitening the data

For step (2), we use PCA on a d-dimensional distribution (the proposed CCILCA algorithm will compute PCA with a slight modification), the unit principal component vectors $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_d$ (eigenvectors of the covariance matrix of $\mathbf{x} \in \mathcal{X}$) are found where $\mathbf{v}_i$ is associated with the eigenvalue λ_i, with $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d$. The space $\mathcal{X}$ is

[3]To see this global versus local contrast more intuitively, consider the problem of shape approximation. The (global) *Taylor expansion* is not practical for approximating a complex real-world shape due to its use of higher order derivatives. In contrast, low order *splines* are more effective.

divided into two parts, the major space M:

$$M = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k\}$$

and the residual space $R = \text{span}\{\mathbf{v}_{k+1}, \mathbf{v}_{k+2}, ..., \mathbf{v}_d\}$ so that $\mathcal{X} = M \oplus R$ where $\oplus$ denotes the orthogonal sum. The dimension k of space M is chosen so that the ratio of the covariance in R over the original space $\mathcal{X}$, $\sum_{i=k+1}^{d} \lambda_i / \sum_{i=1}^{d} \lambda_i$ is a small percentage (e.g., 1%). From the result of PCA, the whitening matrix is:

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_k] \text{diag}\{\lambda_1^{-1/2}, \lambda_2^{-1/2}, ..., \lambda_k^{-1/2}\}.$$

The whitened observation is $\mathbf{y} = \mathbf{V}^\top(\mathbf{x} - \bar{\mathbf{x}})$. In the following, we assume that $\mathbf{y}$ is whitened: it has a zero mean and a unit covariance matrix.

3.2.3 Local: Lobe components

The sample space of a white, zero-mean random vector $\mathbf{y}$ in k-dimensional space can be illustrated by a k-dimensional hypersphere, as shown in Fig. 3.11. It is known that although the whitening process only normalizes the variance along orthogonal principal directions, the whitened vector $\mathbf{y}$ has a unit variance in all the possible directions. That is the meaning of the circle in Fig. 3.11. But the data can go out of the circle.

If the distribution of $\mathbf{y}$ is Gaussian, the samples will be equally dense in all directions. In general, the distribution is not Gaussian and the probability density may

concentrate along certain directions (although the global covariance of projections along any given direction is unit). Each major cluster along a direction is called a lobe, illustrated in Fig. 3.11 as a petal "lobe". Each lobe may have its own fine structure (e.g., sublobes). The shape of a lobe can be of any type, depending on the distribution, not necessarily like the petals in Fig. 3.11. If we assume that $\mathbf{x}$ and $-\mathbf{x}$ are equally likely (e.g., a patch of skin will have the equal chance of feeling the onset and the offset of a press), the distribution is then symmetric about the origin. (But this is not necessarily true in general and, consequently, nonsymmetric lobes can be defined.)

Given a limited resource — c vectors, LCA divides the sample space $\mathcal{X}$ of $\mathbf{x}$ into c mutually nonoverlapping regions, called *lobe regions*:

$$\mathcal{Y} = R_1 \cup R_2 \cup ... \cup R_c, \tag{3.22}$$

where $R_i \cap R_j = \phi$, if $i \neq j$, as illustrated in Fig. 3.11. Each region is represented by a single unit feature vector $\mathbf{w}_i$, called the *lobe component.*

The partition issue presented here is similar to Kohonen's Self Organizing Maps (SOM) (Kohonen, 2001) or, more general, vector quantization (Gray and Neuhoff, 1998; Zador, 1982; Gersho, 1979). We do not use Euclidean distance as with SOM and quantization (we use the concept of high-dimensional probability density of a whitened random vector space instead). Further, each region is infinite (while most cells in SOM and vector quantization are bounded), since we will generate a response $T(\mathbf{x})$ for any vector in the whitened space. $T(\mathbf{x})$ does not approximate $\mathbf{x}$ but represents

it as a feature parameter vector (vector quantization uses the code itself as a direct approximation of input $\mathbf{x}$). The response is useful for visualization, decision-making, prediction or regression.

Suppose we have computed c lobe components (column unit vectors) $\mathbf{w}_i$, $i = 1, 2, ..., c$. They are not necessarily orthogonal and not necessarily linearly independent, depending on the distribution of the whitened space and the number c. They span a lobe feature subspace

$$F = \text{span}\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_c\}. \tag{3.23}$$

Any vector $\mathbf{x}$ in M is represented by the inner product vector:

$$l = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_c]^\top \mathbf{x}.$$

It is called the response from the lobe components, which can be used for visualization or further decision-making using support vector machines (Christianini and Shawe-Taylor, 2000; Lin, 2002) or the Incremental Discriminant Regression developed by Weng & coworkers (Hwang and Weng, 2000; Weng and Hwang, 2002).

The discrete probability in the k-dimensional whitened space M can be estimated in the following way. Each region R_i keeps the number of hits $n(i)$, which records the number of times samples of $\mathbf{x}$ fall into region R_i. Then, the continuous distribution

of $\mathbf{x}$ can be estimated by a discrete probability distribution of c regions:

$$P\{\mathbf{x} \in R_i\} = \frac{n(i)}{n}, \quad i = 1, 2, ..., c.$$

The larger the number c, the more regions can be derived and, thus, the finer partition of the sample space $\mathcal{Y}$.

3.2.4 Belongingness

The remaining major issue is how to compute the lobe regions. We need to determine what kind of vector $\mathbf{w}_i$ is best to represent a lobe region R_i (or for that matter, any subregion of M).

If $\mathbf{x}$ belongs to a region R_i, $\mathbf{x}$ can be approximated by $\mathbf{w}_i$ as $\mathbf{x} \approx \hat{\mathbf{x}} = (\mathbf{x} \cdot \mathbf{w}_i)\mathbf{w}_i$ (it minimizes the mean square error $E\|\mathbf{x} - \hat{\mathbf{x}}\|^2$ for a fixed unit $\mathbf{w}_i$). We determine the unit vector $\mathbf{w}_i$ as the one that minimizes the following error of approximation:

$$\|\mathbf{x} - (\mathbf{x} \cdot \mathbf{w}_i)\mathbf{w}_i\|^2 = \mathbf{x}^\top \mathbf{x} - (\mathbf{x} \cdot \mathbf{w}_i)^2, \tag{3.24}$$

after expansion using the fact that $\mathbf{w}_i$ is a unit vector. The expected error of the above approximation over region R_i is

$$\begin{aligned} E\|\mathbf{x} - (\mathbf{x} \cdot \mathbf{w}_i)\mathbf{w}_i\|^2 &= E[\mathbf{x}^\top \mathbf{x} - (\mathbf{x} \cdot \mathbf{w}_i)^2] = E[\mathbf{x}^\top \mathbf{x} - \mathbf{w}_i^\top \mathbf{x}\mathbf{x}^\top \mathbf{w}_i] \\ &= E[\mathbf{x}^\top \mathbf{x}] - \mathbf{w}_i^\top E[\mathbf{x}\mathbf{x}^\top]\mathbf{w}_i = \text{trace}(\Sigma_\mathbf{x}) - \mathbf{w}_i^\top \Sigma_\mathbf{x} \mathbf{w}_i \end{aligned} \tag{3.25}$$

where $\Sigma_{\mathbf{x}}$ is the covariance matrix of $\mathbf{x}$. We determine the unit $\mathbf{w}_i$ to maximize the above approximation error. Since trace($\Sigma_{\mathbf{x}}$) is constant, the unit $\mathbf{w}_i$ that minimizes the above expression is the one that maximizes $\mathbf{w}_i^\top \Sigma_{\mathbf{x}} \mathbf{w}_i$. From the standard theory of PCA (e.g., see (Jolliffe, 1986)), we know that the solution $\mathbf{w}_i$ is the unit eigenvector of $\Sigma_{\mathbf{x}}$ associated with the largest eigenvalue $\lambda_{i,1}$:

$$\lambda_{i,1}\mathbf{w}_i = \Sigma_{\mathbf{x}}\mathbf{w}_i.$$

In other words, $\mathbf{w}_i$ is the first principal component of $\Sigma_{\mathbf{x}}$, where expectation of $\Sigma_{\mathbf{x}}$ is over the region of approximation, R_i or any other region. Relacing $\Sigma_{\mathbf{x}}$ by the sample covariance matrix, we have

$$\lambda_{i,1}\mathbf{w}_i \approx \frac{1}{n}\sum_{t=1}^{n} \mathbf{x}(t)\mathbf{x}(t)^\top \mathbf{w}_i = \frac{1}{n}\sum_{t=1}^{n} (\mathbf{x}(t)\cdot\mathbf{w}_i)\mathbf{x}(t). \tag{3.26}$$

We can see that the best lobe component vector can be estimated by the average of $\mathbf{x}(t)$ multiplied by the projection $(\mathbf{x}(t)\cdot\mathbf{w}_i)$. This is an important batch estimation equation: It means that the *candid* version of $\mathbf{w}_i$ is equal to the average of $\mathbf{y}_t = (\mathbf{x}(t)\cdot\mathbf{w}_i)\mathbf{x}(t)$. By *candid*, we mean that we keep the covariance of the projections onto $\mathbf{w}_i$ (which is $\lambda_{i,1}$) with $\mathbf{w}_i$ and, thus, the estimator for $\lambda_{i,1}\mathbf{w}_i$ is computed instead of the unit $\mathbf{w}_i$ alone. This scheme is needed for the quasi-optimal efficiency to be discussed in Sec. 3.2.7.

If $\mathbf{w}_i(n-1)$ has already been estimated using $n-1$ samples, we can use it to

compute $\mathbf{y}_t(n)$ defined as:

$$\mathbf{y}_t(n) = \frac{\mathbf{x}(t) \cdot (\mathbf{w}_i(n-1))}{\|\mathbf{w}_i(n-1)\|}\mathbf{x}(t). \tag{3.27}$$

Then Eq. (3.26) states that the lobe component vector is estimated by the average:

$$\lambda_{i,1}\mathbf{w}_i \approx \frac{1}{n}\sum_{t=1}^{n}\mathbf{y}_t(n). \tag{3.28}$$

The larger the inner product $\mathbf{x}(t) \cdot \mathbf{w}_i$, the more applicable the input $\mathbf{x}(t)$ is to the region that the lobe component $\mathbf{w}_i$ represents. Thus, the inner product $\mathbf{x}(t) \cdot \mathbf{w}_i$ is the "belongingness" of $\mathbf{x}(t)$ to the region represented by $\mathbf{w}_i$. For symmetric region R_i, we use the absolute value of $\mathbf{x}(t) \cdot \mathbf{w}_i$ as the belongingness.

This mechanism not only enables us to compute the best $\mathbf{w}_i$ given a region, but also enables many lobe component vectors to compete when data $\mathbf{x}(t)$ are sequentially received. The vector $\mathbf{w}_i$ whose belongingness is the highest is the "winner," which best inhibits all other vectors. The winner uses the current input $\mathbf{x}(t)$ to update its vector, as in Eq. (3.26), but all others do not.

3.2.5 Statistical efficiency

Suppose that there are two estimators Γ_1 and Γ_2, for vector parameter $\theta = (\theta_1, ..., \theta_k)$, which are based on the same set of observations $S = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n\}$. If the expected square error of Γ_1 is smaller than that of Γ_2, i.e., $E\|\Gamma_1 - \theta\|^2 < E\|\Gamma_2 - \theta\|^2$, the estimator Γ_1 is more statistically efficient than Γ_2.

Statistical estimation theory reveals that for many distributions (e.g., Gaussian and exponential distributions), the sample mean is the most efficient estimator of the population mean. This follows directly from Theorem 4.1, p. 429-430 of Lehmann (Lehmann, 1983), which states that under some regularity conditions satisfied by most distributions (such as Gaussian and exponential distributions), the maximum likelihood estimator (MLE) $\hat{\theta}$ of the parameter vector θ is asymptotically efficient, i.e. its asymptotic covariance matrix is the Cramér-Rao information bound (the lower bound) for all unbiased estimators, via convergence in probability to a normal distribution:

$$\sqrt{n}(\hat{\theta} - \theta) \xrightarrow{p} N\{0, I(\theta)^{-1}\} \tag{3.29}$$

in which the Fisher information matrix $I(\theta)$ is the covariance matrix of the score vector $\left\{\frac{\partial f(\mathbf{y},\theta)}{\partial \theta_1}, ..., \frac{\partial f(\mathbf{y},\theta)}{\partial \theta_k}\right\}$, and $f(\mathbf{y}, \theta)$ is the probability density of random vector $\mathbf{y}$ if the true parameter value is θ (see, e.g., Lehmann (Lehmann, 1983), p. 428). The matrix $I(\theta)^{-1}$ is called information bound since under some regularity constraints, any unbiased estimator $\widetilde{\theta}$ of the parameter vector θ satisfies $\mathrm{cov}(\widetilde{\theta} - \theta) \geq I(\theta)^{-1}/n$ (see, e.g., Lehmann (Lehmann, 1983), p.428 or Weng et al. (Weng et al., 1993), pp. 203-204)[4].

Since in many cases the MLE of the population mean is the sample mean, we estimate the mean θ of vector $\mathbf{y}$ by the sample mean. Thus, we estimate an independent vector $\lambda_1 \mathbf{w}_i$ by the sample mean in Eq. (3.28), where $\mathbf{y}_t(n)$ is a "random

[4]For two real symmetric matrices A and B of the same size, $A \geq B$ means that $A - B$ is nonnegative definite, which implies, in particular, that the diagonal elements are all nonnegative, which gives the lower bound for the variance of every element of the vector estimator of θ.

observation".

Since we do not know the distribution of $\mathbf{y}_t(n)$ and it is even dependent on the currently estimated $\mathbf{w}_i$ (i.e., the observations are from a nonstationary process), we use the amnesic mean technique below which gradually "forgets" old "observations" (which use bad $\mathbf{y}_t(n)$ when n is small) while keeping the estimator reasonably efficient.

3.2.6 Amnesic mean

The mean in Eq. (3.28) is a batch method. For incremental estimation, we use what is called an amnesic mean (Weng et al., 2003).

$$\bar{\mathbf{y}}^{(n)} = \frac{n-1-\mu(n)}{n}\bar{\mathbf{y}}^{(n-1)} + \frac{1+\mu(n)}{n}\mathbf{y}_n \tag{3.30}$$

where $\mu(n)$ is the amnesic function depending on n. If $\mu \equiv 0$, the above gives the straight incremental mean. The way to compute a mean incrementally is not new but the way to use the amnesic function of n is new for computing a mean incrementally.

$$\mu(n) = \begin{cases} 0 & \text{if } n \leq n_1, \\ c(n-n_1)/(n_2-n_1) & \text{if } n_1 < n \leq n_2, \\ c + (n-n_2)/r & \text{if } n_2 < n, \end{cases} \tag{3.31}$$

in which, e.g., $c = 2, r = 10000$. As can be seen above, $\mu(n)$ has three intervals. When n is small, the straight incremental average is computed. Then, $\mu(n)$ changes from 0 to 2 linearly in the second interval. Finally, n enters the third section where $\mu(n)$ increases at a rate about $1/r$, meaning the second weight $(1+\mu(n))/n$ in Eq. (3.30)

approaches a constant $1/r$, to slowly trace the slowly changing distribution. Fig. 3.12 shows the development of the amnesic average coefficient, where $w_1 = \frac{n-1-\mu(n)}{n}$ and $w_2 = \frac{1+\mu(n)}{n}$.

For the convenience of description, we write the coefficients of the amnesic average as,

$$\alpha(n) = \frac{n - 1 - \mu(n)}{n}, \tag{3.32}$$

and

$$\beta(n) = \frac{1 + \mu(n)}{n}. \tag{3.33}$$

3.2.7 Efficiency of the amnesic mean

We consider whether the amnesic mean is an unbiased estimator. From the recursive definition of the amnesic mean in Eq. (3.30), we can see that the amnesic mean $\bar{\mathbf{y}}(n)$ is a weighted sum of the involved data $\mathbf{y}_t$:

$$\bar{\mathbf{y}}(n) = \sum_{t=1}^{n} v_t(n)\mathbf{y}_t,$$

where $v_t(n)$ is the weight of data item $\mathbf{y}_t$ which entered at time $t \leq n$ in the amnesic mean $\bar{\mathbf{y}}(n)$. It can be proven using induction on n that the weight $v_t(n)$ is given by the following expression:

$$v_t(n) = \frac{1 + \mu(t)}{t} \prod_{j=t+1}^{n} \frac{j - 1 - \mu(j)}{j}. \tag{3.34}$$

Since all the multiplicative factors above are non-negative, we have $v_t(n) \geq 0$, $t = 1, 2, ..., n$. Using the induction on n, it can be proven that all the weights $v_t(n)$ sum to one for any $n \geq 1$:

$$\sum_{t=1}^{n} v_t(n) = 1. \tag{3.35}$$

(When $n = 1$, we require that $\mu(1) = 0$.) Suppose that the samples $\mathbf{y}_t$ are independently and identically distributed (i.i.d.) with the same distribution as a random variable $\mathbf{y}$. Then, the amnesic mean is an unbiased estimator of $E\{\mathbf{y}\}$:

$$E\{\bar{\mathbf{y}}(n)\} = E\{\sum_{t=1}^{n} v_t(n)\mathbf{y}_t\} = \sum_{t=1}^{n} v_t(n)E\{\mathbf{y}_t\} = \sum_{t=1}^{n} v_t(n)E\{\mathbf{y}\} = E\{\mathbf{y}\}.$$

Let $\mathrm{cov}(\mathbf{y})$ denote the covariance matrix of $\mathbf{y}$. The expected mean square error of the amnesic mean $\bar{\mathbf{y}}(n)$ is

$$\mathrm{cov}(\bar{\mathbf{y}}(n)) = \left(\sum_{t=1}^{n} v_t^2(n)\right) \mathrm{cov}(\mathbf{y}) = c(n)\mathrm{cov}(\mathbf{y}). \tag{3.36}$$

where we define the *error coefficient*:

$$c(n) = \sum_{t=1}^{n} v_t^2(n).$$

When $\mu(t) = 0$ for all n, the error coefficient becomes $c(n) = 1/n$ and Eq. (3.36) returns to the expected square error of the regular sample mean:

$$\mathrm{cov}(\bar{\mathbf{y}}(n)) = \frac{1}{n}\mathrm{cov}(\mathbf{y}). \tag{3.37}$$

It is then expected that the amnesic mean for a stationary process will not have the same efficiency as the straight sample mean for a stationary process. Fig. 3.13 shows the error coefficient $c(n)$ for three different amnesic functions $\mu(n) \equiv 2$, $\mu(n) \equiv 1$ and $\mu(n) \equiv 0$. The smaller the error coefficient, the smaller the expected square error, but also less capability to adapt to a changing distribution. The three cases shown in Fig. 3.13 indicate that when $n > 10$, the amnesic mean with $\mu(t) \equiv 1$ increased about 50% (for the same n) from that for $\mu(t) \equiv 0$ and with $\mu(t) \equiv 2$ it increased about 100%.

From Fig. 3.13 we can see that a constant positive $\mu(t)$ is not best when t is small. The multi-sectional amnesic function $\mu(t)$ in Eq. (3.31) performs a straight average for small t to reduce the error coefficient for earlier estimates, and when t is very large, the amnesic function changes with t to track the slowly changing distribution. Therefore, the multi-sectional amnesic function $\mu(t)$ is more suited for practical signals with unknown nonstationary statistics. It is appropriate to note that the exactly optimality of the multisectional amnesic function is unlikely under an unknown nonstationary process (not i.i.d.), unless an assumption of certain types of nonstationary process is imposed, which is not, however, necessarily true in reality.

In summary, we should not expect that an estimator suited for an unknown nonstationary process to have the same expected efficiency as ones for an i.i.d. stationary process. The distribution of signals received in many applications is typically nonstationary and, therefore, an amnesic mean with a multisectional (dynamic) amnesic function is better (see later Fig. 3.16).

3.2.8 Proposed CCILCA Algorithm

The Candid Covariance-free Incremental LCA algorithm incrementally computes the lobe matrix

$$\mathbf{W}(t) = [\mathbf{w}_1^{(t)}, \mathbf{w}_2^{(t)}, ..., \mathbf{w}_c^{(t)}],$$

for lobes, from whitened samples $\mathbf{x}(1), \mathbf{x}(2), ...$ of dimension k without computing the $k \times k$ covariance matrix. The number c is the number of lobe components to be extracted (pre-determined limited resource). The length of $\mathbf{w}_i$ is the variance of projections of the vectors $\mathbf{x}(t)$ in the i-th region onto $\mathbf{w}_i$. The algorithm is shown in Algorithm 3.

Algorithm 3 The CCILCA algorithm.

1: Initialize using observations: for $t = 1, 2, ..., c$, $\mathbf{w}_t^{(c)} = \mathbf{x}(t)$ and $n_t = 1$.
2: **for** $t = c+1, c+2, ...$ **do**
3: **for** $i = 1, 2, ..., c$ **do**
4: Compute the response of all lobe vectors

$$l_i = \mathbf{x}(t) \cdot \mathbf{w}_i^{(t-1)} / \|\mathbf{w}_i^{(t-1)}\|.$$

5: Decide the winner: $j = \arg\max_{1 \le i \le c}\{|l_i|\}$.
6: Update the number of wins $n_j \leftarrow n_j + 1$, compute the amnesic weight coefficient $\alpha(n_j + 1)$ and $\beta(n_j + 1)$with Eq. 3.32 and Eq. 3.33.
7: update winner using amnesic mean of the winner lobe component:

$$\mathbf{w}_j^{(n_j)} = \alpha(n_j + 1)\mathbf{w}_j^{(n_j - 1)} + \beta(n_j + 1) l_j \mathbf{x}(t) \tag{3.38}$$

8: **end for**
9: **end for**

Algorithm 3 requires that input be whitened. The following is the CCILCA algorithm that does not require inputs to be whitened. It normalizes the output power of each lobe component while computing all the lobe components.

Algorithm 4 CCILCA algorithm without pre-whitening.

Initialize using observations: $n = 0$. For $t = 1, 2, ..., c$, do $\mathbf{w}_t^{(c)} = \mathbf{x}(t)$, $n_t = 0$, and $\sigma_t = \|\mathbf{x}(t)\|^2$.

for $t = c+1, c+2, ...$ **do**

For $i = 1, 2, ...c$, compute the projection of the i-th cell

$$l_i = \mathbf{x}(t) \cdot \mathbf{w}_i^{(t-1)} / \|\mathbf{w}_i^{(t-1)}\|.$$

Generate the normalized response $z_i = l_i / \sqrt{\sigma_i}$.

Decide the winner among the normalized responses:

$$j = \arg \max_{1 \le i \le c} \{|z_i|\}.$$

For the winner, the j-th cell, $n_j \leftarrow n_j + 1$, compute the amnesic weight coefficient $\alpha(n_j + 1)$ and $\beta(n_j + 1)$with Eq. 3.32 and Eq. 3.33.

Update the amnesic average σ_j (the variance of response) of all cells:

$$\sigma_j \leftarrow \alpha(n_j + 1)\sigma_j + \beta(n_j + 1)l_j^2.$$

Update the winner's lobe component vector using the amnesic average (cell's Hebian learning):

Update the weights for the winner:

$$\mathbf{w}_j^{(t)} = \alpha(n_j + 1)\mathbf{w}_j^{(t-1)} + \beta(n_j + 1)z_j\mathbf{x}(t)/\sqrt{\sigma_j},$$

end for

The above two algorithms are simple. They never compute any $k \times k$ matrix (covariance-free).

The above algorithm computes CCI PCA (Weng et al., 2003) if the step 2 is cut short to contain only steps (a) and (c). Step (c) is modified so that for each vector $\mathbf{w}_i$, the residual from the previous vector $\mathbf{w}_{i-1}^{(t)}$ is computed, and then the residual is used as the input to the next vector $\mathbf{w}_i^{(t-1)}$ for updating. No winner-take-all is needed.

Therefore, the pre-whitening step can be computed by the same type (CCI) of algorithm. The difference in the pre-whitening step is to use the residual so that the resulting vectors are orthogonal (a requirement of PCA).

After we computed the whitening matrix $\mathbf{V}$ and the lobe component matrix $\mathbf{W}$, we have $\mathbf{B}^\top = \mathbf{W}^\top \mathbf{V}^\top$ and $l = \mathbf{B}^\top(\mathbf{y} - \bar{\mathbf{y}})$ as projections onto lobe components, or response of lobe components.

We can also define a neighborhood function $h_{ji}(t)$, which acts as a smoothing kernel. Usually, $h_{ji}(t)) = h(\|r_j - r_i\|, t)$, where r_j, r_i are the location vectors of node j and i. $h_{ji}(t)$ often has the shape of "Mexican hat" or triangle. Then, the algorithm is changed to Algorithm 5.

3.2.9 Time and space complexities

Given each k-dimensional input $\mathbf{x}$, the time complexity for updating c lobe components and computing all the responses from $\mathbf{x}$ is $O(ck)$. Since LCA is meant to run in real-time, this low update complexity that is important. If there are t input vectors,

Algorithm 5 Topographic CCILCA.

1: $\mathbf{w}_i = \mathbf{y}_i$, $i = 0, 1, 2, ..., k$.
2: $n_i = 1$, $i = 1, 2, ..., k$.
3: **for** $t = 1, 2, ...$ **do**
4: $\quad j = \arg\max_i \left\{ \frac{\left|\mathbf{w}_i(n_i)^\top \cdot \mathbf{y}_t\right|}{\|\mathbf{w}_i(n_i)\|} \right\}$
5: $\quad$ **for** All component vector $\mathbf{w}_i(n_i)$, $i = 1, 2, ..., k$ **do**
6:

$$\mathbf{w}_i(n_i + 1) = \alpha(n_i)\mathbf{w}_i(n_i) + h_{ji}(t)\beta(n_i)\frac{\mathbf{w}_i(n_i)^\top \cdot \mathbf{y}_t}{\|\mathbf{w}_i(n_i)\|}\mathbf{y}_t, \tag{3.39}$$

where $\mathbf{w}_i(n_i)$ is the component vector $\mathbf{w}_i$ after the n_i-th updating, $\alpha(n_i)$ and $\beta(n_i)$ are given in Eqs. 3.32 and 3.33, respectively.
7: $\quad n_i = n_i + 1$.
8: $\quad$ **end for**
9: **end for**

the total amount of computation is $O(ckt)$.

The LCA's space complexity is $O(ckt)$, for c neurons with k dimensional input $\mathbf{x}$.

In fact, the above space and time complexities are the **lowest possible** ones: Since c vectors need to be computed and each vector has k components, the space complexity cannot be lower than $O(ck)$. Further, the time complexity cannot be lower than $O(ckt)$ because the responses $l = (l_1, l_2, ..., l_c)$ for each of t inputs need that many computations.

Suppose that each lobe component (vector) is considered as a "neuron" and the number of hits $n(j)$ is its clock of "maturity," which determines the single weight w_1 ($w_2 = 1 - w_1$) for its updating. The CCILCA algorithm is an *in-place development* algorithm, in the sense that the network does not need extra storage or an extra developer. The winner-take-all mechanism is a computer simulation of the lateral inhibition mechanism in the biological neural networks (see, e.g., Kandel et al. (Kandel et al., 2000) p. 4623). The inhibition-winner updating rule is a computer simulation

of the Hebbian mechanism in the biological neural networks (see, e.g., Kandel et al (Kandel et al., 2000) p.1262.).

3.2.10 LCA is ICA for super-Gaussians

Further, the components that have a super-Gaussian distribution correspond to lobe components we defined here. Each linear combination of k super-Gaussian independent components correspond to symetric lobes, illustrated in Fig. 3.11. Therefore, if components in $s(t)$ are all super-Gaussian, finding lobe components by CCILCA is equivalent to finding independent components.

3.3 Experiment Results

3.3.1 Low dimensional simulation

The proposed algorithm was first applied to a low dimensional data set, which contains 2 i.i.d source components. Each of the source components is taken from a Laplace distribution of zero mean and unit variance, thus, is a super-Gaussian random variable. Fig. 3.14(a) shows 5000 sample points used in the experiment. The directions of the source independent components and evaluated independent components are illustrated, which are almost identical. A more detailed error over the number of samples used is shown in Fig. 3.14(b).

3.3.2 Comparison with Other ICA Methods

Suppose that the distribution of k-dimensional random input $\mathbf{x}$ is stationary. In CCILCA, the lobe component vector $\mathbf{w}_i$ converges to the eigenvalue scaled eigenvector $\lambda_{i,1}h_{i,1}$ in the mean square sense and the speed of convergence is estimated as

$$E_i\|\mathbf{w}_i^{(t)} - \lambda_{i,1}h_{i,1}\|^2 \approx \frac{2}{t}k\sigma$$

where σ is the estimated average component-wise variance of observation $\mathbf{y}_t = (\mathbf{x}(t) \cdot \mathbf{w}_i^{(t-1)})\mathbf{x}(t)/\|\mathbf{w}_i^{(t-1)}\|$. The convergence is not to a local extremum. It is to the correct solution. The quasi-optimal convergence speed is due to the use of statistical efficiency in the algorithm design. If the distribution of $\mathbf{x}$ changes slowly, the above error estimate is still applicable.

The quasi-optimal statistical efficiency appears to drastically improve the capacity to deal with high-dimensional data. We selected two state-of-the-art incremental ICA algorithms, the Type-2 Extended Bell-Sejnowski (ExtBS) (Lee et al., 1999) and Type-3 (NPCA-LS) (Pajunen and Karhunen, 1998; Karhunen et al., 1998) for performance comparison with our proposed Type-4 CCILCA algorithm. The choice of these two algorithms is due to their superior performance in the comparison results of (Giannakopoulos et al., 1998). We used the downloaded code from the authors for the ExtBS algorithm.

The NPCA algorithm used was proposed in (Pajunen and Karhunen, 1998) with $\beta = 0.998$ and g as tanh. The ExtBS algorithm was run with the following set of parameters: blocksize = 1, learning rate = 0.001, learning factor = 0.985,

momentum constant = 0.1, number of iterations = 1, step = 5, and block size for kurtosis estimation is 1000. This version is called ExtBS1, the number 1 indicating that the block size for updating was 1. Thus, ExtBS 1 is a partial sequential algorithm, sequential for independent component update but computation for kurtosis is block-incremental. We called it Type-2.5.

To investigate the capabilities to handle high-dimensional data, we synthetically generated random data with different dimensions. The number n of samples used in each run is a function of the input dimension d (we used $n = 1000d$ samples). Each independent source has a Laplacian distribution. The mixing matrix was chosen randomly and was non-degenerate. The error between the direction of the true and the estimated independent components was measured as the angle between them in radians. The results were averaged over 50 runs. Figs. 3.15(a) and (b), show that these two better performing ICA algorithms have problems in converging when the dimension d increases beyond 5 and the correct convergence fails when $d = 25$. The CCILCA consistently give considerably smaller error and take the least CPU time among the three. To show more detail about the profile of decreasing error, we plot the errors in Fig. 3.15(c). As indicated by the figure, both the Type-3 algorithm NPCA and Type-2.5 ExtBS1 do not converge for a moderate dimension of $d = 25$ although ExtBS1 does fine when $d = 2$. The proposed Type-4 CCILCA does well.

Next, we compare our CCILCA algorithm with Type-2 Extended Bell-Sejnowski (or extended infomax) (Lee et al., 1999) with block size 1000 and the Type-1 batch algorithm FastICA (Hyvärinen and Oja, 1997; Hyvärinen and Oja, 2000). This is *not a fair* comparison since a Type-4 algorithm (like CCILCA) should not be expected to

out-perform a Type-1 or Type-2 algorithm. We compare them anyway to understand the limit when CCILCA is compared with two state-of-the-art Type-1 and Type-2 algorithms.

It is well known that ICA algorithms are "data grizzlies". Typically, even for a low dimension simulation task (e.g., $d = 2$), ICA algorithms need thousands of samples to approach the independent components. Convergence in respect to the number of samples used in training is a good evaluation of the efficiency of ICA algorithms.

For a higher dimension, we synthetically generated random observations from an i.i.d. Laplacian random vector with dimension of 100. The results are shown in Fig. 3.16, where the y-axis marks the number of samples and the x-axis indicates the average error in radians. In order to show more detailed aspects of CCILCA, three variations are tested. "LCA with fixed m" (m stands for the amnesic function $\mu(n)$) and "LCA with dynamic m" are original LCA methods with a fixed μ and varying $\mu(n)$ as defined in Eq. (3.31), respectively. "LCA eliminating cells" algorithm dynamically eliminates cells whose hitting rate is smaller than 3/4 of the average hitting rate, since sometimes two vectors share a single lobe (which is very rare and does not significantly affect the purpose of density estimation by lobe components but does affect our error measure). As shown in Fig. 3.16, all the three Type-4 LCA algorithms converged very fast, faster than the Type-2 algorithm Extended infomax and even the Type-1 FastICA. The batch Extended Infomax algorithm needs much more samples, therefore, it did not converge in these tests.

It is somewhat **surprising** that the proposed Type-4 CCILCA algorithm, operating under the most restrictive condition, out-performs the state-of-the-art Type-3,

Type-2, and Type-1 algorithms by a remarkably wide margin. We think this is mainly due to the new lobe component concept and the quasi-optimal property of the statistical efficiency. The simple structure of the CCILCA algorithm is also very attractive.

3.3.3 Cocktail Party Problem

We test the algorithm on a simulation of the cocktail party problem. Nine sound sources are mixed by an orthogonal matrix. Each sound source is 6.25 seconds long and the sampling rate is 8.0KHz in 8 bits mono format. Therefore each sound source contains 50000 values. The sound clips are downloaded from `http://www.cis.hut.fi/projects/ica/cocktail/cocktail_en.cgi`, where they use these sound clips to test the FastICA algorithm (Hyvärinen, 2001).

Here, we simplified the problem since the mixing matrix is a pure rotation matrix,

thus no whitening process is needed. The mixing matrix is shown in Eq. 3.40.

$$\mathbf{M} = \begin{bmatrix} -0.47 & 0.31 & -0.30 & -0.07 & 0.13 & 0.27 & 0.55 & -0.07 & -0.43 \\ -0.22 & -0.17 & -0.11 & 0.36 & 0.14 & 0.24 & -0.65 & 0.05 & -0.53 \\ -0.52 & -0.18 & -0.21 & 0.08 & -0.66 & 0.21 & -0.07 & 0.11 & 0.39 \\ 0.41 & -0.18 & -0.02 & 0.41 & 0.05 & 0.71 & 0.25 & -0.18 & 0.18 \\ 0.43 & -0.26 & -0.11 & -0.29 & -0.57 & 0.03 & 0.12 & 0.14 & -0.54 \\ -0.26 & -0.28 & 0.62 & 0.12 & -0.17 & -0.11 & 0.16 & -0.40 & -0.20 \\ 0.18 & 0.24 & -0.45 & 0.55 & -0.23 & -0.46 & 0.05 & -0.37 & -0.04 \\ 0.05 & 0.06 & -0.32 & -0.54 & 0.03 & 0.20 & -0.34 & -0.66 & 0.11 \\ -0.11 & -0.78 & -0.39 & -0.04 & 0.34 & -0.24 & 0.22 & -0.02 & 0.06 \end{bmatrix} \tag{3.40}$$

Fig. 3.17(a) shows one of the nine original source signals. Fig. 3.17(b) displays one of the nine mixed sound signals. The mixed signals are first whitened, then we applied the proposed algorithm to the mixed sound signals. It is worth noting that the proposed algorithm is an incremental method. Therefore, unlike other batch ICA method doing iterations on the date set, we have used data only once and then discarded them. In Fig. 3.17(c) The independent components quickly converge to the true ones, with a good approximation as early as 1.5 second.

3.3.4 Natural Image ICA

As one of our major applications, we also conduct our experiment on natural image patches. 500,000 of 16 × 16-pixel image patches are randomly taken from thirteen

natural images, available from http://www.cis.hut.fi/projects/ica/imageica/. The CCILCA algorithm is applied to the pre-whitened image patches $k = 16 \times 16 = 256$ to update the lobe component matrix V. The matrix $(B^{\top})^{-1}$ is then computed. Each column of the matrix is shown in Fig. 3.18(a) by a 16×16 patch, as the features of the natural scenes. A total of 256 256-dimensional vectors are displayed in the figure. They all look smooth and most of them are localized (only a small patch are non-zero, or gray) as expected. The entire process take less than 46 minutes (i.e., 181 frames/second) on a Pentium III 700MHz PC with 512MB memory compared to over 10 hours of learning time for the FastICA algorithm using 24% of the 500,000 samples (disk thrashing is also a factor).

Fig. 3.18(b) shows how many times each lobe component was the top "winner". Most components have roughly a similar rate of hits, except relatively few leftmost (top) ones and rightmost (tailing) ones. Although it is not exactly true that each lobe component is equally likely to be hit, nearly equal hits for the majority is a desirable property for a high-dimensional density estimation due to the following consideration:

Suppose that a random number z can take a discrete number of values $z = z_i$, with $P\{z = z_i\} = p_i$, $i = 1, 2, ..., c$. Given a fixed c, the discrete distribution that maximizes the entropy is a uniform distribution (Papoulis, 1991): $p_i = 1/c$, $i = 1, 2, ..., c$. In other words, the distribution of hits by lobe components has nearly the largest entropy.

We apply the topological LCA algorithm, shown in Algorithm 5, to natural image set as well. Fig. 3.19 displays the filters of the topographic CCILCA algorithm. Filters show iso-orientation preference in a neighborhood. And the orientation blocks

are surrounding a singular position at row No.9 and column No.8. This is resembling the pin-wheel structure found in the biological visual cortex. The winning times are shown in Fig. 3.20. The neuron at the singular position get significantly more hits than other neurons.

3.4 Derivation of the LCA algorithm

In this section we discuss the mathematical analysis of the LCA algorithm from a sparse optimization point of view. Here we start from a intuitive sparseness measurement explanation, and end up with the LCA learning algorithm.

3.4.1 ℓ^p norm sparseness measurement

As we mentioned in the section 3.1.6, sparseness is an important property of super-Gaussian random variable. Maximize sparseness is a desirable strategy that is used by different ICA algorithms. But the measurement of sparseness is still illusive. Many heuristic measurements of sparseness have been proposed, e.g. Olshausen and Field proposed the following form:

$$Sp(\mathbf{u}) = -\sum_i S(u_i), \tag{3.41}$$

where $Sp(\mathbf{u})$ is defined as the sparseness of random vector $\mathbf{u}$; $S(u_i)$ is a nonlinear function, and u_i is the response of the neuron. Olshausen and Field suggest $S(u_i)$ to be an even nonnegative function, e.g. $-e^{-u^2}$, $\log\left(1+u^2\right)$, and $|u|$. Intuitively,

when there are more neurons (random vector $\mathbf{u}$) firing at the same time, the larger the function $\sum_i S(u_i)$ would be. Therefore, maximizing Eq. 3.41 will maximize the sparseness.

Karvanen and Cichocki (Karvanen and Cichocki, 2003) generalized the measure of sparseness to ℓ^p norm criteria:

$$Sp(\mathbf{u}) = E\left\{\left(\sum_i \left|u_i^p\right|\right)^{\frac{1}{p}}\right\}. \tag{3.42}$$

It is clear that Olshausen et. al's sparseness measure is a special case of ℓ^p norm, when p is equal to 1 and $|u|$ is chosen as the S function. When $p = 4$, the ℓ^p norm is related to the widely used non-Gaussianality measure kurtosis [5].

Karvanen et. al suggest the range of p in $(0, 1]$, and particularly smaller p, such as $p = 0.1$ or $p = 0.01$, should be used. *However we found when $p \to \infty$, it will lead to a surprisingly good independent component analysis algorithm.* Let us first take look at an example. Fig. 3.21 shows the joint distribution of two Laplace random variables. The original components are mixed with a rotation matrix of 20 degree. We then projected the data set the a series of orthogonal basis, which is a rotated from 0 degree to 180 degree with 1 degree interval. After projected to the basis, mean of different ℓ^p norms are calculated and plotted. Fig. 3.22 displays the mean of ℓ^p norm of different projections. It is interesting to see the ℓ^2 norm is a straight line, which makes sense, because rotation will not change the Euclidean distance. Surprisingly,

[5]The kurtosis is defined as $kurt(u) = E\{u^4\} - 3$, where u has unit variance.

the optima of ℓ^p norm curve are inverted on the two side of the ℓ^2 norm, which means the maxima of the ℓ^p norm with $p < 2$ are corresponding to the positions of the minima of ℓ^p norm with $p < 2$. Then to find the independent component (in this case it is along the 20 degree and 110 degree directions), one needs to find the minima of the ℓ^p norm with $p < 2$ or maxima of ℓ^p norm with $p > 2$.

Another issue is robustness of the estimation when noise presents. From Fig. 3.22, we can see almost all the norm curves agree on the same optima position (except ℓ^2 norm), but the norm curve with greater difference between the maxima and minima will be more robust when noise is presented. So the choice of p can be either close to zero or close to infinity. That is why Karvanen et. al suggested smaller p, such as 0.1 or 0.01. The problem of small p is that the optimization process is difficult, since the gradient of such criteria typically involves inverting matrices.

On the other hand, at first glance $p \to \infty$ is not a good choice either, but since it generally holds that:

$$\lim_{p\to\infty} \left(\sum_i |u_i|^p \right)^{\frac{1}{p}} = \max\{|u_i|\}. \tag{3.43}$$

So it leads to a surprisingly simple computation. In the next section we will also see the gradient of infinity norm also has a simple form, thus the optimization process is relatively easy for the infinity norm case. Therefore, we can define the following criteria function:

$$J(\mathbf{u}) = E\left\{ \lim_{p\to\infty} \left(\sum_i |u_i|^p \right)^{\frac{1}{p}} \right\} = E\left\{ \max_i \{u_i\} \right\}, \tag{3.44}$$

and maximizing this function solves the maximizing sparseness problem.

3.4.2 Infinity norm sparseness optimization

As we defined in Eq. 3.44 in the last section, this criteria function that will maximize the mean of ℓ^∞ *norm* can also be written as:

$$J(\mathbf{W}) = \int \max_j \left[\left(\mathbf{w}_j^\top \mathbf{y} \right)^2 \right] p(\mathbf{y})\, d\mathbf{y}, \tag{3.45}$$

where we replace the $|u_j|$ with $(\mathbf{w}_j^\top \mathbf{y})^2$ for convenience, since the absolute function is not differentiable . So the goal is to maximize this function with respect to $\mathbf{W}$.

Intuitively saying, if $\mathbf{w}_j$ is along the tail direction of the multivariate super-Gaussian distributions, the criteria function will be maximized. So that our goal is to maximize the Eq. 3.45 with respect to the $\mathbf{w}_j$.

The closed-form for the determination of $\mathbf{w}_j$ is not available with the general random variable $\mathbf{y}$ with pdf $p(\mathbf{y})$. So we have to resort to an iterative approximation scheme.

The difficulty of the analysis lies in the discontinuity caused by the maximum function, $\max_j \left[\left(\mathbf{w}_j^\top \mathbf{y} \right)^2 \right]$. This problem can be circumvented by going back to the infinity norm form as shown in Eq. 3.43.

Let $c = \arg\max_j \left[\left(\mathbf{w}_j^\top \mathbf{y} \right)^2 \right]$. Thus, the maximum part can be rewritten as:

$$\begin{aligned}\left(\mathbf{w}_c^\top \mathbf{y}\right)^2 &= \max_j \left[\left(\mathbf{w}_j^\top \mathbf{y}\right)^2\right] \\ &= \lim_{r\to\infty} \left[\sum_j \left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r}\right]^{\frac{1}{r}} \end{aligned} \tag{3.46}$$

To use the gradient descent approach to find the maximum of the Eq. 3.45, we take the partial derivative of Eq. 3.45 with respect to $\mathbf{w}_j$:

$$\frac{\partial E}{\partial \mathbf{w}_j} = \int \lim_{r\to\infty} \frac{\partial \left\{ \left[\sum_j \left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r}\right]^{\frac{1}{r}} \right\}}{\partial \mathbf{w}_j} p(\mathbf{y})\, d\mathbf{y}. \tag{3.47}$$

Let

$$A = \sum_j \left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r}. \tag{3.48}$$

Eq. 3.46 can be written as $\lim_{r\to\infty} A^{\frac{1}{r}}$. Since,

$$\begin{aligned}\frac{\partial A}{\partial \mathbf{w}_j} &= 2r \sum_i \left[\left(\mathbf{w}_i^\top \mathbf{y}\right)^{2r-1} \cdot \mathbf{y} \cdot \partial \mathbf{w}_i \Big/ \partial \mathbf{w}_j\right], \\ &= 2r \left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r-1} \mathbf{y}, \end{aligned} \tag{3.49}$$

then we have:

$$\begin{aligned}\frac{\partial A^{\frac{1}{r}}}{\partial \mathbf{w}_j} &= \frac{1}{r} A^{\frac{1}{r}-1} \cdot \frac{\partial A}{\partial \mathbf{w}_j} \\ &= 2A^{\frac{1}{r}-1} \left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r-1} \mathbf{y}. \end{aligned} \tag{3.50}$$

Denote

$$
\begin{aligned}
B &= \frac{\left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r-1}}{A} \\
&= \frac{\left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r-1}}{\sum_i \left(\mathbf{w}_i^\top \mathbf{y}\right)^{2r}} \\
&= \left[\sum_i \frac{\left(\mathbf{w}_i^\top \mathbf{y}\right)^{2r}}{\left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r}}\right]^{-1} \left(\mathbf{w}_j^\top \mathbf{y}\right)^{-1} \\
&= \left[\frac{\left(\mathbf{w}_1^\top \mathbf{y}\right)^{2r}}{\left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r}} + \ldots + \frac{\left(\mathbf{w}_c^\top \mathbf{y}\right)^{2r}}{\left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r}} + \ldots \frac{\left(\mathbf{w}_n^\top \mathbf{y}\right)^{2r}}{\left(\mathbf{w}_j^\top \mathbf{y}\right)^{2r}}\right]^{-1} \left(\mathbf{w}_j^\top \mathbf{y}\right)^{-1}.
\end{aligned}
\tag{3.51}
$$

Given $\mathbf{w}_j$, the term $\frac{\left(\mathbf{w}_c^\top \mathbf{y}\right)}{\left(\mathbf{w}_j^\top \mathbf{y}\right)}$ is the maximum. When $r \to \infty$, the term $\left(\frac{\mathbf{w}_c^\top \mathbf{y}}{\mathbf{w}_j^\top \mathbf{y}}\right)^{2r}$ will dominate the series of summation. So,

$$
\begin{aligned}
\lim_{r\to\infty} B &= \lim_{r\to\infty} \left(\frac{\mathbf{w}_c^\top \mathbf{y}}{\mathbf{w}_j^\top \mathbf{y}}\right)^{-2r} \left(\mathbf{w}_j^\top \mathbf{y}\right)^{-1} \\
&= \delta_{cj} \left(\mathbf{w}_j^\top \mathbf{y}\right)^{-1},
\end{aligned}
$$

where δ_{cj} is the Kronecker delta. It has the following form:

$$
\delta_{cj} = \begin{cases} 1 & \text{if } c = j, \\ 0 & \text{otherwise.} \end{cases}
$$

Substitute B into Eq. 3.50, we obtain

$$\begin{aligned}\lim_{r\to\infty}\frac{\partial A^{\frac{1}{r}}}{\partial \mathbf{w}_j} &= 2\left(\mathbf{w}_c^{\top}\mathbf{y}\right)^2\delta_{cj}\left(\mathbf{w}_j^{\top}\mathbf{y}\right)^{-1}\mathbf{y} \\ &= 2\delta_{cj}\left(\mathbf{w}_c^{\top}\mathbf{y}\right)\mathbf{y} \end{aligned} \tag{3.52}$$

The Eq. 3.50 is also the updating rule for the weight matrix $\mathbf{W}$. With the consideration of the most efficient estimation, we can come up with the updating rule exactly the same as Eq. 3.38.

3.5 Broader Impact

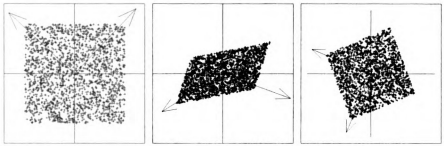
In neuroscience and machine perception, there is a growing interest in simulating the developmental process of feature detectors in the early sensory pathways (see, e.g., Elman, Bates, Johnson, Karmiloff-Smith, Parsi & Plunkett (Elman et al., 1997); Weng, McClelland, Pentland, Sporns, Stockman, Sur & Thelen (Weng et al., 2001); Weng & Stockman (Weng and Stockman, 2002)). In such sensory pathways, there is a need for developing feature detectors for all areas (receptive fields), across different positions and sizes, in the sensor array (e.g., camera CCD array or the retina). The total number of such areas is so large (sampled in the y-x positions and in the size range) that it is impractical for each feature detector of k connections to have extra storage space of $k \times k$ for its development. This raises the critical need for Type-4 incremental ICA algorithms.

We will place the Type-4 algorithms developed in this proposed project on the

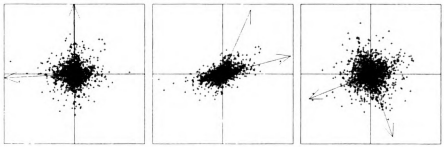
web, freely available to everybody. Due to its simple structure, lowest possible order of time and space complexities, quasi-optimal statistical efficiency, and the Type-4 nature, we expect that this class of algorithms will be widely used.

3.6 Conclusion

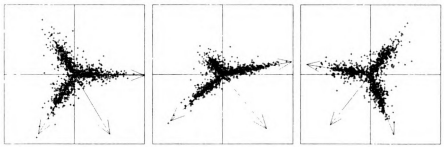
The proposed CCILCA algorithm is simple and fast under multiple demanding computational requirements: high dimensional, incremental and free of higher order statistics computation. This is due to the simplicity of the winner-take-all principle and the most efficient estimation concept used in the algorithm design. The sparse representation can come from a winner-take-all competitive learning neural network, which helps to explain some learning mechanism in the neural system.



Factorizable into two independent uniform distributions (sub-Gaussians)



Factorizable into two independent Laplacian distributions (super-Gaussians)



Nonfactorizable distribution

Figure 3.10: Illustration of lobe components. The plus signs are random samples. 1st column: the original source signal. 2nd column: the observed mixed signals (after affine transform from the left). 3rd column: whitened signals. Lobe components are marked by arrows.

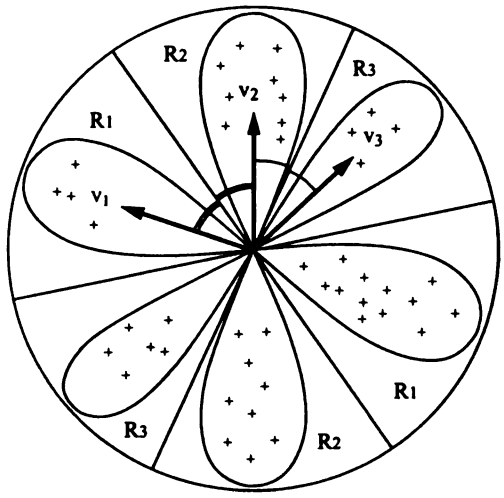


Figure 3.11: The sample space of a zero-mean white random vector $\mathbf{x}$ in 2-D space can be illustrated by a circle. Each mark + indicates a random sample of $\mathbf{x}$. The distribution is partitioned into $c = 3$ (symmetric) lobe regions R_i, $i = 1, 2, 3$, where R_i is represented by the lobe component (vector) $\mathbf{w}_i$.

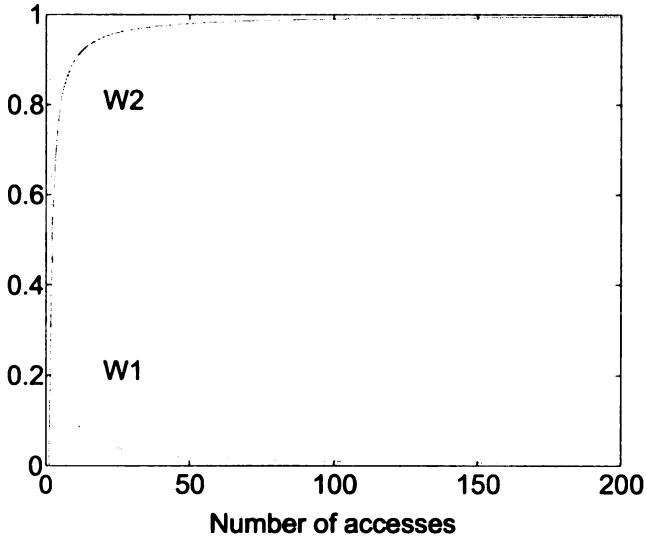


Figure 3.12: The amnesic average coefficient. X axis is the number of visits. Y axis is the weight coefficient of w_1 and w_2.

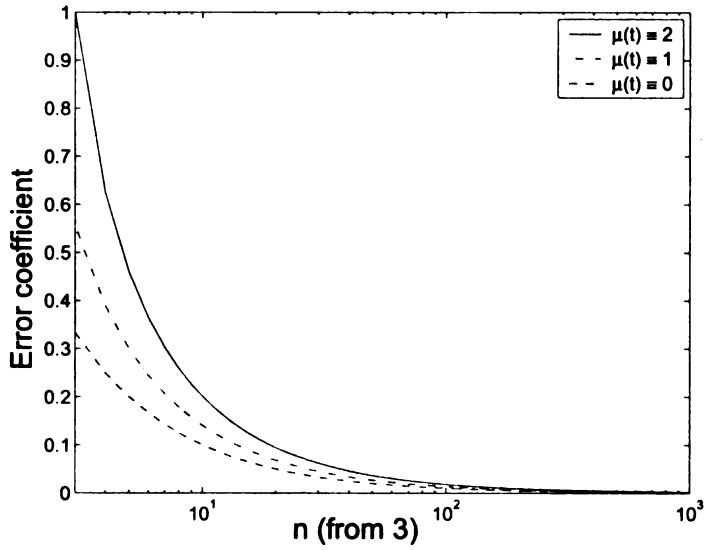


Figure 3.13: The error coefficients $c(n)$ for amnesic means with different amnesic functions $\mu(n)$.

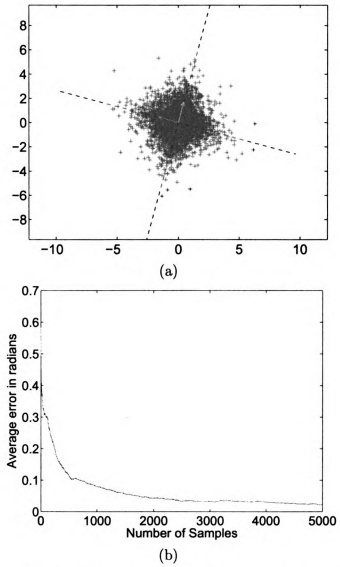
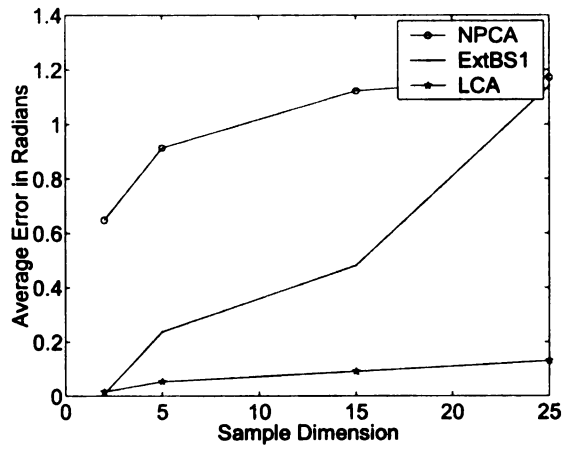
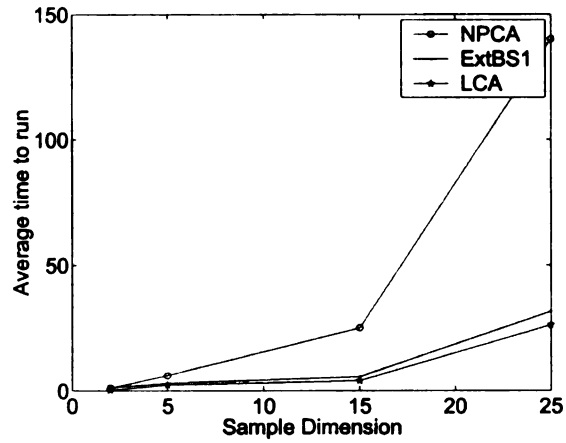


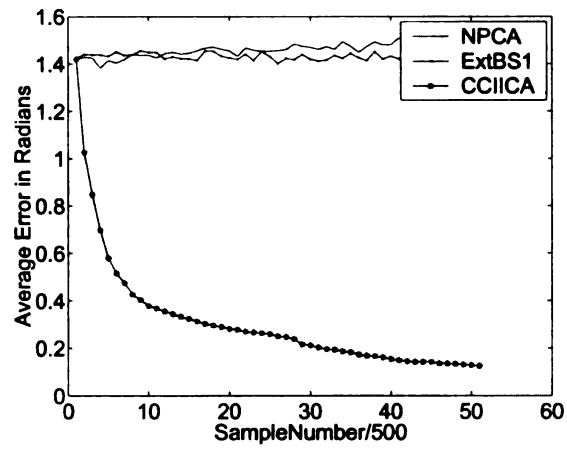
Figure 3.14: The proposed method on 2-D Laplace distribution data set. (a) The source independent components and the evaluated independent components. The dash-dot lines indicate the direction of source independent components, and the two arrows indicate the components evaluated by the proposed method. (b) Average error in the angle of the evaluated independent components vectors and the source independent component vectors.



(a)



(b)



(c)

Figure 3.15: Comparison of ICA results among (Type-3) NPCA, (Type-2.5) ExtBS1 and (Type-4) CCILCA for super-Gaussian sources. (a) Average error for different data dimension d. (b) Average time to run for different data dimension d. (c) Average error for $d = 25$ as a function of the number of samples.

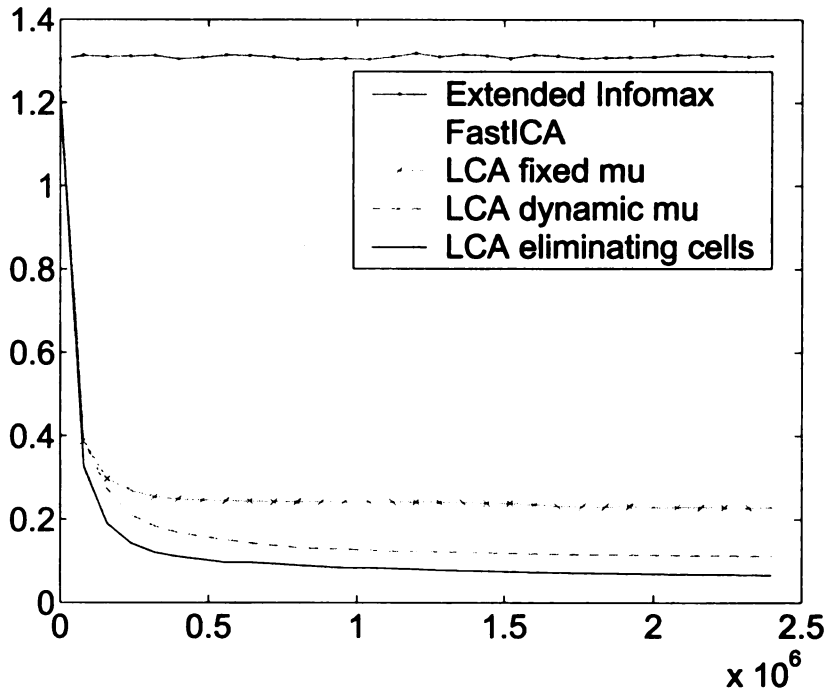
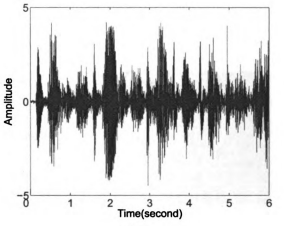
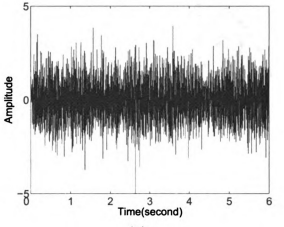


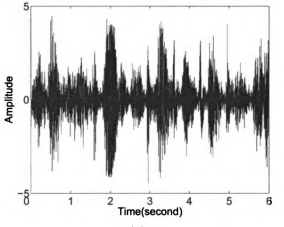
Figure 3.16: Comparison among (Type-2) Extended Infomax, (Type-1) FastICA and three variants of the proposed Type-4 CCILCA algorithms.



(a)

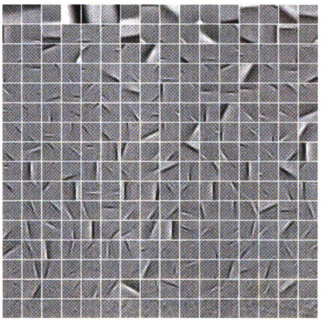


(b)

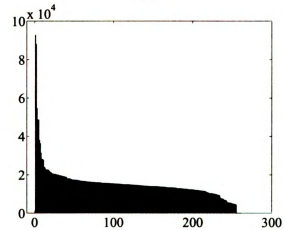


(c)

Figure 3.17: Cocktail party problem. (a) A music sound clip in its original form. It is one of the nine sound sources. (b) One of the nine mixed sound signals. (c) Recovered music sound wave. Comparing to (a), the sound signal is recovered after approximately 1.5 second.



(a)



(b)

Figure 3.18: Lobe components from natural images (not factorizable). (a) LCA derived features from natural images, ordered by the number of hits in decreasing order. (b) The numbers of hits of the corresponding lobe components in (a).

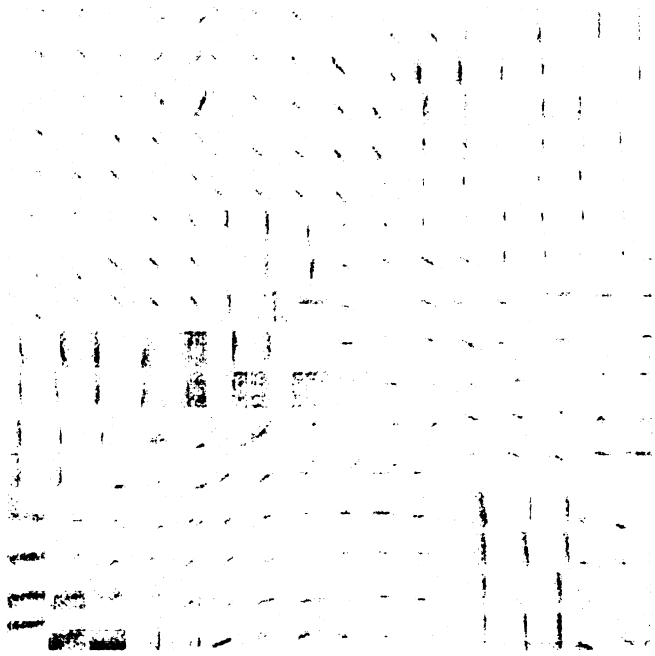


Figure 3.19: Results of Algorithm 5. Topographically ordered basis functions in ICA of natural images.

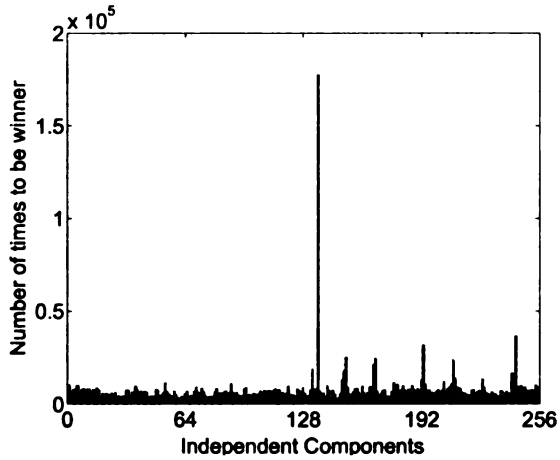


Figure 3.20: Results of Algorithm 5. Number of times for each independent component to be the "winner." The noticeable big hit corresponding to the neuron at row No.9 and column No. 8.

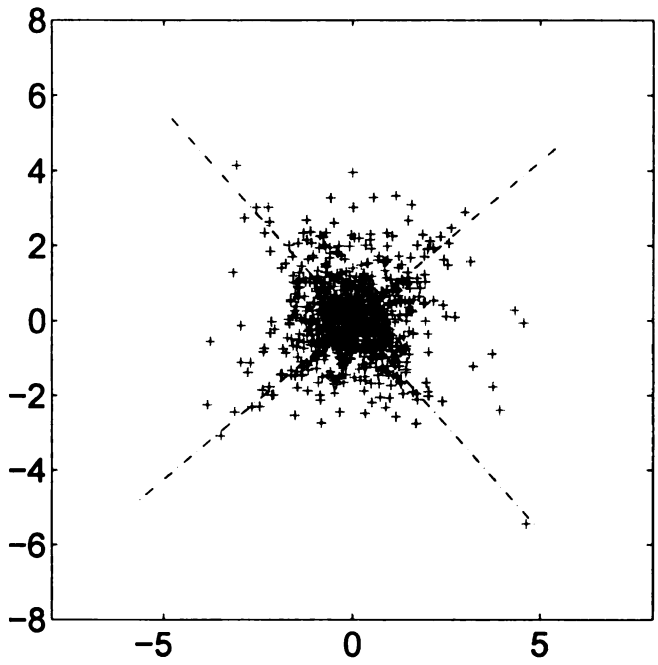


Figure 3.21: Data set is joint distribution of two unit variance Laplace distribution. The two independent Laplace random variable are mixed by a rotation matrix of 20 degree.

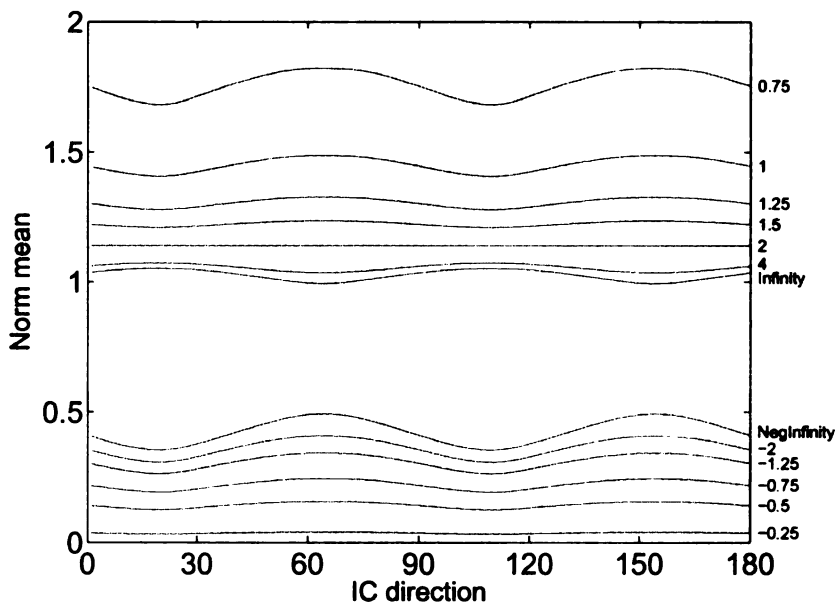


Figure 3.22: The ℓ^p norm criteria functions under different p. The data set is projected onto a series of orthogonal basis, and then the expectation of ℓ^p norms of all the projected samples are computed.

Chapter 4

Sparse representation and Feature Selection

In the chapter 3 Lobe Component Analysis algorithm based on a novel sparseness measure is discussed. In this chapter we further discuss the sparseness optimization. We propose a new perspective to achieve sparseness via the winner-take-all principle for the linear kernel regression and classification task. We form the duality of the Least Absolute Shrinkage Selection Operator (LASSO) (Tibshirani, 1996) criteria, and transfer an ℓ_1 norm minimization to an ℓ_∞ norm maximization problem. Two solutions are proposed: it can be solved by standard quadratic programming with linear inequality constraints. We show that the number of parameters to be estimated in the ℓ_∞ normed space is half of the parameters in the solution suggested by David Gay for ℓ_1 normed space. Second, we introduce a novel winner-take-all neural network solution derived from gradient descending, which links the sparse representation and the competitive learning scheme. This scheme is a form of unsupervised learning in

which each input pattern comes through learning, to be associated with the activity of one or at most a few neurons. However, the lateral interaction between neurons in the same layer is strictly preemptive in this model. This framework is applicable to a variety of problems, such as Independent Component Analysis (ICA), feature selection, and data clustering.

4.1 Introduction: The Supervised Learning Problem

The central problem of supervised learning or regression can be formulated as a function approximation. In either case we have pairwise correspondence of samples $\mathbf{x_i}$ and y_i, where $\mathbf{x_i} \in \mathcal{R}^d$, $y_i \in \mathcal{R}$, and d is the dimension of the input space. Especially, when y_i is discrete, e.g. $y_i \in \{-1, 1\}$, the problem is called *classification*, whereas continuous y_i (for example $y_i \in \mathcal{R}$) indicates a *regression* problem. The task is to infer a function $f(\cdot)$, such that $\mathbf{y} = f(\mathbf{x})$. More precisely, if the model of the function is chosen, the function can be written as $\mathbf{y} = f(\mathbf{x}, \beta)$, where β is the parameter vector of the model, $\mathbf{y} = [y_1, ...y_n]^\top$, and n is number of training samples. The model of the function $f(\cdot)$ can have various forms. For example, in a three-layer perceptron neural networks (Bishop, 1995), the function has the following form,

$$\mathbf{y} = f_2(\mathbf{W}_2 f_1(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_0) + \mathbf{b}_1), \tag{4.1}$$

where $f_i(\cdot)$ is a non-linear limit function, like *sigmoid*. Then $\beta = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_0, \mathbf{b}_1\}$ is the parameter set to be optimized.

In this study, we consider the form of the function to be linear with respect to the parameter vector *beta*, i.e.,

$$f(\mathbf{x}, \beta) = \sum_{i=1}^{k} \beta_i h_i(\mathbf{x}) = \beta^\top \mathbf{h}(\mathbf{x}), \tag{4.2}$$

where $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), ...h_k(\mathbf{x})]^\top$ is a vector of k determined functions of the input vector $\mathbf{x}$. Typical settings of this function are:

1. Linear regression, where $\mathbf{h}(\mathbf{x}) = [1, x_1, ..., x_d]^\top$, and d is the dimension of the input space. Apparently, $k = d + 1$.

2. Non-linear regression, where $\mathbf{h}(\mathbf{x}) = [\phi_1(\mathbf{x}), ..., \phi_\mathbf{k}(\mathbf{x})]^\top$. If $\phi_i = g(\mathbf{W}\mathbf{x})$, where $g(\cdot)$ is the sigmoid function, this model is the single layer perceptron. Note that usually $\phi_1(\mathbf{x}) = \mathbf{1}$.

3. Kernel regression, where $\mathbf{h}(\mathbf{x}) = [1, K(\mathbf{x}, \mathbf{x}_1), ...K(\mathbf{x}, \mathbf{x}_{k-1})]^\top$. $K(\mathbf{x}, \mathbf{x}_i)$ is some kernel function, e.g. Gaussian kernels.

Typically, it is assumed that the output variable $\mathbf{y}$ from the training set is contaminated by additive white Gaussian noise, *i.e.* $\mathbf{y} = f(\mathbf{x}, \beta) + \mathbf{w}$, where $\mathbf{w} = [w_1, ..., w_n]^\top$ is a set of i.i.d. white Gaussian random variables with variance σ^2. Thus, the conditional probability $p(\mathbf{y}|\beta)$ is Gaussian, *i.e.*

$$p(\mathbf{y}|\beta) = \mathcal{N}(\mathbf{y}|\mathbf{h}(\mathbf{x})^\top \beta, \sigma^2 \mathbf{I}), \tag{4.3}$$

where $\mathbf{I}$ is the identity matrix. We write $\mathbf{H} = \mathbf{h}(\mathbf{x})^\top$, and $\mathbf{H}$ is called design matrix. The element (i, j) of $\mathbf{H}$, denoted by H_{ij}, is given by $H_{ij} = h_j(x_i)$.

Since the noise is assumed to be i.i.d, by simply applying the Maximum Likelihood Estimator (MLE), we obtain the log-likelihood,

$$l(\beta) = \ln p(\mathbf{y}|\beta) = -\frac{\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{k} H_{ij}\beta_j\right)^2}{\sigma^2} = -\frac{\|\mathbf{y} - \mathbf{H}\beta\|_2^2}{\sigma^2} \tag{4.4}$$

where $\|\cdot\|_2$ is the ℓ_2 norm, i.e. Euclidian norm of the vector, which is defined by $\|\mathbf{z}\|_2 = \sqrt{\mathbf{z}^\top \mathbf{z}}$. The estimated parameter vector $\hat{\beta}$

$$\hat{\beta} = (\mathbf{H}^\top \mathbf{H})^{-1}\mathbf{H}^\top \mathbf{y}. \tag{4.5}$$

The problem of the least square error estimation is that it emphasizes the fitting accuracy, however with limited data samples it usually leads to a severe over-fitting problem. That is the trained function works well on the training samples set, but it generates large error on the testing set. This drawback is often referred as low generalization ability.

Note in this case there is no preference on the parameter vector β, so its prior is of a uniform distribution. This prior typically leads to some large values of the coefficient $beta_i$, and is considered to have more complexity and roughness.

With a zero-mean Gaussian prior for β with diagonalized covariance matrix A, the estimation can be formalized by *maximum a posteriori* (MAP) process. The prior of β then becomes an ℓ_2-norm regularization term in the log-posteriori, where it will

prefer a smaller β. When $A = \mu^2 I$, it is called *ridge regression* (Hoerl and Kennard, 1970). The posteriori is given by

$$\begin{aligned} l_{MAP}(\beta) &= \ln\left(p(\mathbf{y}|\beta)p(\beta|a)\right) \\ &= -\frac{\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{k} H_{ij}\beta_j\right)^2}{\sigma^2} - \frac{\sum_{i=1}^{k}\beta_i^2}{\mu^2} \\ &= -\frac{\|\mathbf{y} - \mathbf{H}\beta\|_2^2}{\sigma^2} - \frac{\|\beta\|_2^2}{\mu^2} \end{aligned}$$

It is worth noting that for a Gaussian prior, the solution $\hat{\beta}_{ridge}$ of the MAP and Bayesian estimation agrees at the same point and has the following close form solution,

$$\hat{\beta}_{ridge} = \left(\sigma^2\mathbf{A}^{-1} + \mathbf{H}^\top\mathbf{H}\right)^{-1}\mathbf{H}^\top\mathbf{y}.$$

However, for other priors the MAP and Bayesian estimation generally have different solution, and analytical solution may not exist.

The Ridge regression has the similar over-fitting problem as the MLE, since the regulization term in the objective function favors smaller and smooth coefficients in β, but does not penalize a complex model, which means almost all the coefficients in β are non-zero and one needs to keep all the basis function in the design matrix $\mathbf{H}$. In contrast to this limitation, a sparse estimation of β is defined as one, in which most of the coefficients in β are zero with only a few non-zero elements. Sparseness is desirable in the ways that

1. it leads to a simpler model, since most of the basis function in design matrix $\mathbf{H}$ will not be used.

2. the generalization ability improves with a sparse model.

3. sparseness corresponds to performing feature and variable selection.

If sparseness is preferred, then a Laplacian prior can be adopted for β, *i.e.*

$$p(\beta|\alpha) = \left(\frac{\alpha}{2}\right)^k \exp(-\alpha \|\beta\|_1),$$

where α is a parameter of the Laplacian pdf, and $\|\mathbf{x}\|_1$, $i = 0, ...\infty$ is the so called ℓ_1-norm, which equals to $\|\beta\|_1 = \sum_{i=1}^{k} |\beta_i|$. The Laplacian distribution features a heavy tail and has a high concentration at the near-zero area. It means that most of the β's components will be zero, and the probability of having a large value is relatively high, compared to the Gaussian distribution with the same variance. Fig. 3.2 compares three prior distributions, i.e. Laplacian, Gaussian, and Uniform.

Utilizing the same MAP process, the estimation of β is given by

$$\hat{\beta} = \arg\min_{\beta} \left\{ \|\mathbf{y} - \mathbf{H}\beta\|_2^2 + t \|\beta\|_1 \right\}, \tag{4.6}$$

where $t = 2\sigma^2\alpha$ is a control parameter, which can favor either the squared error term or the ℓ_1-norm regularization term. This criterion is also known as Least Absolute Shrinkage Selection Operator (LASSO) (Tibshirani, 1996). It is worth noting here that due to the non-Gaussian prior, the MAP estimation is not equivalent to the Bayesian estimation as it is in the ridge regression. Therefore, the estimation is bi-

ased. To make a unbiased estimation, one needs to integrate in all β space, which is computationally prohibitive. However, if the posterior concentrates at a certain point, then this biased estimation may only have a small variance from the unbiased estimation, which is desirable. This is one reason why a concentrated sparse prior is preferred. Some researchers introduced "hyper-parameter" to further steepen the prior, for example, in Relevance Vector Machine (RVM) (Tipping, 2001) and Figueiredo's work (Figueiredo, 2003).

Another reason that a sparse representation is desirable is because it improves the generalization of a learning system. For example, in supervised learning, the goal is to infer a mapping based on the training samples. The generalization capability is accomplished by reducing the complexity of the model, which is characterized by the number of non-zero parameters. This problem is formalized as ℓ_0-norm minimization. It is known that ℓ_0-norm minimization is NP-hard (Amaldi and Kann, 1998). However, it is established that the solution of the ℓ_1 problem is the same as the ℓ_0 problem if a certain condition is satisfied. So, ℓ_1-norm problem is still an important issue.

In this chapter, we propose a method to solve the ℓ_1-norm version of the problem. We construct the dual problem of LASSO criterion as in Eq. (4.6) and use a gradient-based method to find the solution. This method is by no means claimed to be superior to the quadratic programming (QP) based method; however, it opens a perspective to address the problem differently. We have shown that the proposed method actually applies the competitive learning principle. In addition, this method can also motivate other biologically plausible models for solving similar problems.

The reminder of the chapter is organized as follows: In Section 4.2 we formulate the dual problem of the LASSO, and propose a solution based on gradient descent. We reformulate the proposed algorithm in Section 4.2.5. The experiment results and comparison with existing sparse optimization algorithms is in Section 4.3. Section 4.4 provides conclusions.

4.2 Method

4.2.1 Duality

First, we will construct the dual problem of Eq. (4.6). Fig. 4.1 illustrates this problem. The Eq. (4.6) can be geometrically explained as the minimum ℓ_1-norm between the origin and the convex set K. This distance, according to duality theory, is equal to the maximum ℓ_∞-norm distance between the origin and the plane that separates the origin and the convex set K. In Fig. 4.1, the parabola denotes the convex set K. Point B is the closest point to the origin, in terms of the ℓ_1-norm. Actually, the ℓ_1-norm of B consists of two parts, the quadratic form of Eq. (4.6) and $t\,\|\beta\|_1$ in Eq. (4.6), as illustrated in Fig. 4.1.

Point A is on the tangent plane of point B, and A is the closest point to the origin, in terms of the ℓ_∞-norm. The duality theory establishes that the ℓ_∞ norm reaches its maximum value while the ℓ_1-norm reaches its minimum value. Thus, with a few manipulations, we get the following theorem to formulate this intuitive geometric explanation.

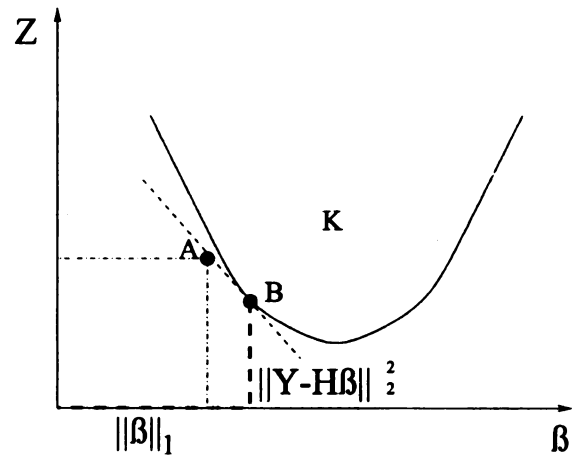


Figure 4.1: Geometry explanation of duality. Point B is the closest point in K to the origin.

Theorem 4.2.1 *If we have*

$$\hat{\beta}' = \arg\max_{\beta} - \frac{\begin{bmatrix} \frac{\partial Z}{\partial t\beta} \\ -1 \end{bmatrix}^{\top} \begin{bmatrix} t\beta \\ Z \end{bmatrix}}{\left\| \begin{bmatrix} \frac{\partial Z}{\partial t\beta} \\ -1 \end{bmatrix} \right\|_{\infty}}$$

where $Z = \|\mathbf{y} - \mathbf{H}\beta\|_2^2$, *then* $\hat{\beta}' = \hat{\beta}$, *and* $\hat{\beta}$ *is the same as in Eq.* (4.6).

The generalized proof can be found in (Luenverger, 1969).

4.2.2 Derivations

Let $Z = \|\mathbf{y} - \mathbf{H}\beta\|_2^2 = \beta^{\top}\mathbf{H}^{\top}\mathbf{H}\beta - 2\mathbf{y}^{\top}\mathbf{H}\beta + \mathbf{y}^{\top}\mathbf{y}$, and denote

$$J(\beta) = \begin{bmatrix} \frac{\partial Z}{\partial t\beta} \\ -1 \end{bmatrix}^{\top} = \begin{bmatrix} 2/t(\mathbf{H}^{\top}\mathbf{H}\beta - \mathbf{H}^{\top}\mathbf{y}) \\ -1 \end{bmatrix}^{\top} \tag{4.7}$$

Now we can formulate the dual problem of the original LASSO criterion,

$$E(\beta) = -\frac{J(\beta)\begin{bmatrix} t\beta \\ Z \end{bmatrix}}{\|J(\beta)\|_{\infty}}, \tag{4.8}$$

Therefore,

$$\frac{\partial Z}{\partial t\beta} = \frac{2}{t}(\mathbf{H}^{\top}\mathbf{H}\beta - \mathbf{H}^{\top}\mathbf{y}). \tag{4.9}$$

$$\frac{\partial J}{\partial \beta} = \begin{bmatrix} & & 0 \\ \frac{2}{t}\mathbf{H}^{\top}\mathbf{H} & & \vdots \\ & & 0 \end{bmatrix}. \tag{4.10}$$

Letting $\mathbf{C} = \mathbf{H}^{\top}\mathbf{H} = [\mathbf{c}_1, ..., \mathbf{c}_n]$, and substituting with Eq. (4.10) and Eq. (4.9), we get

$$\frac{\partial E}{\partial \beta} = \frac{2\mathbf{C}\beta}{\|J(\beta)\|_{\infty}} - \frac{\left[\beta^{\top}\mathbf{C}\beta - \mathbf{y}^{\top}\mathbf{y}\right]}{\|J(\beta)\|_{\infty}^2} \cdot \frac{\partial \|J(\beta)\|_{\infty}}{\partial \beta}.$$

Now we proceed to compute the partial derivative of $\|J(\beta)\|_{\infty}$ w.r.t. β.

$$\begin{aligned} \|J(\beta)\|_{\infty} &= \max_i \left\{ \left| \frac{2}{t}\left(\beta^{\top}\mathbf{c}_i - \mathbf{y}^{\top}\mathbf{h}_i\right) \right|, 1 \right\} \\ &= \sqrt{\max_i \left\{ \left| \frac{2}{t}\left(\beta^{\top}\mathbf{c}_i - \mathbf{y}^{\top}\mathbf{h}_i\right) \right|^2, 1 \right\}} \end{aligned}$$

So,

$$\frac{\partial \|J(\beta)\|_\infty}{\partial \beta} = \frac{1}{2}\|J(\beta)\|_\infty^{-1} \cdot \frac{\partial}{\partial \beta}\max_i \left\{ \left| \frac{2}{t}\left(\beta^\top \mathbf{c}_i - \mathbf{y}^\top \mathbf{h}_i\right)\right|^2, 1 \right\}. \tag{4.11}$$

The last partial derivative term in Eq. (4.11) needs some special treatment. The difficulty of the analysis lies in the discontinuity caused by the maximum function. This problem can be circumvented by the use of the following equality. Let $\{a_i\}$ be a set of positive real scalars; then it generally holds that

$$\max_i\{a_i\} \equiv \lim_{r\to\infty}\left[\sum_i a_i^r\right]^{\frac{1}{r}}.$$

This is just another identity of the ℓ_∞-norm, which is differentiable. The proof of the derivative of maximum function can be found in the appendix. In fact, a similar technique can be seen in (Kohonen, 2001), in which Kohonen derived the vector quantization (VQ) algorithm based on this idea. In (Weng and Zhang, 2004), a competitive learning algorithm has been derived from a maximum criteria function. Therefore, based on Eq. B.5, this leads to the following updating rules.

$$\frac{\partial}{\partial \beta}\max_i \left\{ \left| \frac{2}{t}\left(\beta^\top \mathbf{c}_i - \mathbf{y}^\top \mathbf{h}_i\right)\right|^2, 1 \right\} = \begin{cases} \frac{8}{t}\left(\beta^\top \mathbf{c}_m - \mathbf{y}^\top \mathbf{h}_m\right)\cdot \mathbf{c}_m, \text{if } \frac{4}{t^2}\left(\beta^\top \mathbf{c}_m - \mathbf{y}^\top \mathbf{h}_m\right)^2 > 1 \\ [0, 0 \ldots 0]^\top, \text{otherwise.} \end{cases}, \tag{4.12}$$

where

$$m = \arg\max_j \left(\left| \beta^\top \mathbf{c}_j - \mathbf{y}^\top \mathbf{h}_j \right| \right). \tag{4.13}$$

Rearranging these equations, we have the final updating rules,

$$\nabla\beta \propto \begin{cases} \frac{2\mathbf{C}\beta}{\|J(\beta)\|_\infty} - \frac{4\left[\beta^\top \mathbf{C}\beta - \mathbf{y}^\top \mathbf{y}\right]}{t\|J(\beta)\|_\infty^3} \cdot \left(\beta^\top \mathbf{c}_m - \mathbf{y}^\top \mathbf{h}_m\right) \mathbf{c}_m \\ \frac{2\mathbf{C}\beta}{\|J(\beta)\|_\infty} \end{cases} \tag{4.14}$$

The switching condition of these two updating rules is the same as that in Eq. (4.12)

Remarks: Eq. (4.13) defines a competitive learning process. The algorithm will find the winner and use the updating rule with the winner's value.

The aforementioned method will search for the optimal value of β to minimize the criteria function in Eq. (4.6). With the ℓ_1-norm constraint, some of the components β_i of vector β will gradually decay. However, due to the nature of the gradient method, they will not be exactly zero. So an additional step that sets a hard threshold to make smallβ_i to zero will help the convergence of the algorithm.

The choice of learning rate in any gradient algorithm is important. Here we used a dynamic learning rate that is introduced in chapter 3, i.e. amnesic average. Suppose there are two statistical estimators Γ_1 and Γ_2 for estimating parameter θ. If $E\|\Gamma_1 - \theta\|^2 < E\|\Gamma_2 - \theta\|^2$, Γ_1 is said to be more statistically efficient than Γ_2.

We consider the right hand side of Eq. 4.14 as an "observation." The goal is to get the mean of this observation, while β is estimated incrementally. It is known that for many distributions, the sample mean is the most efficient estimator for the mean of the random variable. When the distribution is unknown, the sample mean

is the best linear estimator, which results in the minimum error variance. For many distributions, the sample mean reaches of approaches the Cramér-Rao bound (CRB).

Then, an efficient estimator is one that has the least variance from the real parameter β, and its variance is bounded below by the CRB. Thus, we estimate a set of independent component vectors $\mathbf{w}_i$ by the sample mean of the observation in Eq. 4.14.

The algorithm is guided by the statistical efficiency, but it is not absolutely the most efficient one, because

1. the true distribution of the observation is unknown;

2. the distribution changes with β being incrementally estimated;

3. amnesic average is used to gradually discount "old" observations, which reduces the statistical efficiency moderately.

4.2.3 Quadratic Programming

Quadratic programming (QP) is used for finding the solution of LASSO regression. The method suggested by David Gay (Tibshirani, 1996) transforms the LASSO to a QP problem with $2n$ variables and $(2n+1)$ constraints, where n denotes the number of the kernels. In this subsection, we are proposing the dual-QP algorithm, which converts the original problem into a problem with fewer variables (n) and $2n$ constraints.

The dual problem of LASSO regression is described in Eq. (4.8) and we recapitu-

late it as finding $\hat{\beta}$, s.t.

$$\hat{\beta} = \arg\max_{\beta} -\frac{J(\beta)\begin{bmatrix} t\beta \\ Z \end{bmatrix}}{\|J(\beta)\|_\infty} \tag{4.15}$$

This optimization problem is equivalent to

$$\hat{\beta} = \arg\max_{\beta} -J(\beta)\begin{bmatrix} t\beta \\ Z \end{bmatrix}$$

$$= \arg\min_{\beta} -[t^2\beta^\top(\beta\mathbf{H}^\top\beta\mathbf{H})\beta - \mathbf{y}^\top\mathbf{y}]$$

s.t.

$$\|J(\beta)\|_\infty = 1 \tag{4.16}$$

The constraint Eq. (4.16) can be written as

$$\max(\|\frac{2}{t}\mathbf{H}^\top\mathbf{H}\beta - \mathbf{H}^\top\mathbf{y}\|_\infty, 1) = 1$$

and is simplified to be

$$\|\frac{2}{t}\mathbf{H}^\top\mathbf{H}\beta - \mathbf{H}^\top\mathbf{y}\|_\infty \leq 1 \tag{4.17}$$

Written in component form, Eq. (4.17) becomes

$$\max_i \mid \frac{2}{t}\mathbf{c}_i^\top\beta - \alpha_i \mid \leq 1, \text{ for } i = 1, ..., n$$

where

$$\mathbf{H}^\top\mathbf{H} = \begin{bmatrix} \mathbf{c}_1^\top \\ \mathbf{c}_2^\top \\ \vdots \\ \mathbf{c}_n^\top \end{bmatrix} \text{ and } 2\mathbf{H}^\top\mathbf{y} = \begin{bmatrix} \alpha_1^\top \\ \alpha_2^\top \\ \vdots \\ \alpha_n^\top \end{bmatrix}$$

The inequality can be expressed in

$$-1 \leq \frac{2}{t}\mathbf{c}_i^\top\beta - \alpha_i \leq 1$$

We write the inequality in vector form

$$2\mathbf{H}^\top\mathbf{y} - 1 \leq 2\mathbf{H}^\top\mathbf{H}\beta/t \leq 1 + 2\mathbf{H}^\top\mathbf{y}$$

Thus, the equivalent QP problem becomes

$$\hat{\beta} = \arg\min[\beta^\top\mathbf{H}^\top\mathbf{H}\beta]$$

s.t.,

$$-2\mathbf{H}^\top\mathbf{H}\beta/t \leq 1 - 2\mathbf{H}^\top\mathbf{y}$$

$$2\mathbf{H}^\top\mathbf{H}\beta/t \leq 1 + 2\mathbf{H}^\top\mathbf{y}$$

It is worth noting that the above QP has n variables and $2n$ constraints.

4.2.4 Filter Development

Let us reconsider the original linear kernel regression problem in a special case. Suppose the design matrix $\mathbf{H}$ is an orthogonal n by n matrix, and there is no additive noise. So, $\beta = \mathbf{H}^\top \mathbf{y}$. Also, we are not considering optimization with respect to β, but instead, with respect to the design matrix $\mathbf{H}$. Then, we get the criteria function adapted from Eq. (4.6)

$$\hat{\mathbf{H}} = \arg\min_{\mathbf{H}} \{ \left\| \mathbf{H}^\top \mathbf{y} \right\|_1 \}.$$

Using the same gradient technique described in section 4.2.2, we can get the updating rule of design matrix $\mathbf{H}$,

$$\nabla \mathbf{h}_j \propto 2\delta_{cj}(\mathbf{h}_c^\top \mathbf{y})\mathbf{y},$$

where $c = \arg\max_i \left\{ \frac{|\mathbf{h}_i^\top \cdot \mathbf{y}|}{\|\mathbf{h}_i\|} \right\}$, and δ_{cj} is the Kronecker delta: $\delta_{cj} = \{1 \text{ if } c = j, 0 \text{ otherwise}\}$. This is precisely a "winner-take-all" algorithm. Only the winning kernels (neurons) will get updated. For more details about this method, see (Zhang and Weng, 2004).

4.2.5 Gradient Sparseness Optimization Algorithm

We summarize the training procedure in Algorithm 6. Each iteration of this algorithm is computationally efficient, because it only involves matrix multiplication and maximum function. The time complexity of each iteration is $O(kn)$, where k is the number of basis vectors and m is the dimension of each basis vector.

Algorithm 6 Gradient Sparseness Optimization Algorithm

1: Preprocessing: Get the training set $(\mathbf{x}_i, \mathbf{y}_i)$, $i = 0, 1, 2, ..., n$. For each i, $\mathbf{y}_i = \mathbf{y}_i - \bar{\mathbf{y}}$, where $\bar{\mathbf{y}} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{y}_i$.

2: $\beta_i = 0$, $i = 0, 1, 2, ..., k$.

3: Initialize the design matrix $\mathbf{H}$, and compute $\mathbf{C} = \mathbf{H}^\top \mathbf{H}$.

4: **repeat**

5: $k = 1$.

6: Compute $m = \arg\max_j \left(\left| \beta^\top \mathbf{c}_j - \mathbf{y}^\top \mathbf{h}_j \right| \right)$.

7: **if** $\frac{4}{t^2} \left(\beta^\top \mathbf{c}_m - \mathbf{y}^\top \mathbf{h}_m \right)^2 > 1$ **then**

8: Update β with:

$$\beta_{\text{new}} = w_1 \beta_{\text{old}} + w_2 [\frac{2\mathbf{C}\beta}{\|J(\beta)\|_\infty} - \frac{4\left[\beta^\top \mathbf{C}\beta - \mathbf{y}^\top \mathbf{y}\right]}{t\|J(\beta)\|_\infty^3} \cdot \left(\beta^\top \mathbf{c}_m - \mathbf{y}^\top \mathbf{h}_m \right) \mathbf{c}_m],$$

where w_1 and w_2 is the same as α and β respectively defined in Eq. (3.32) and Eq. (3.33).

9: **else**

10: Update β with:

$$\beta_{\text{new}} = w_1(k) \beta_{\text{old}} + w_2(k) \frac{2\mathbf{C}\beta}{\|J(\beta)\|_\infty}$$

11: **end if**

12: $k = k + 1$.

13: **until** Objective function in Eq. (4.8) reaches the target value, or $\Delta\beta$ is less than some threshold.

4.3 Experiments

4.3.1 Kernel Regression

Our first experiment illustrates the performance of the proposed algorithm in kernel regression. The regression model is

$$\mathbf{y} = f(\mathbf{x}, \beta) = \beta_0 + \sum_{i=1}^{k} \beta_i K(\mathbf{x}, \mathbf{x}_i),$$

where $K(\mathbf{x}, \mathbf{x}_i) = exp\{-\frac{\|\mathbf{x}-\mathbf{x}_i\|^2}{2\sigma^2}\}$ is the kernel function, $\mathbf{x}_i$ and σ are the kernel parameters. The function to be approximated is $1-d$ sinc function $y = \sin(x)/x$. We randomly collected 150 samples and added Gaussian noise with variance 0.01. The first row in Fig. 4.2 shows the fitting results of proposed method, ridge regression, and the proposed dual-QP LASSO regression algorithm, respectively. The dots are samples with noise, and the dashed lines are the ground truth sinc function. Solid lines show the approximation results. The circled dots correspond to the kernels with nonzero weights, a.k.a the "supporting kernels". In the second row, we use the bar figure to display the weights of those kernels. As Fig. 4.2 clearly indicates, both proposed gradient sparseness method and dual-QP LASSO achieve sparseness. The ℓ_∞-norms are also marked on these figures. The proposed method in our testing performs better than LASSO regression.

Fig. 4.3 shows the convergence procedure. The proposed method takes about 200 iterations to converge. Fig. 4.4 illustrates how the control parameter $t = 2\sigma^2\alpha$ affects the mean square error and model sparseness. We conducted 20 tests with t ranging

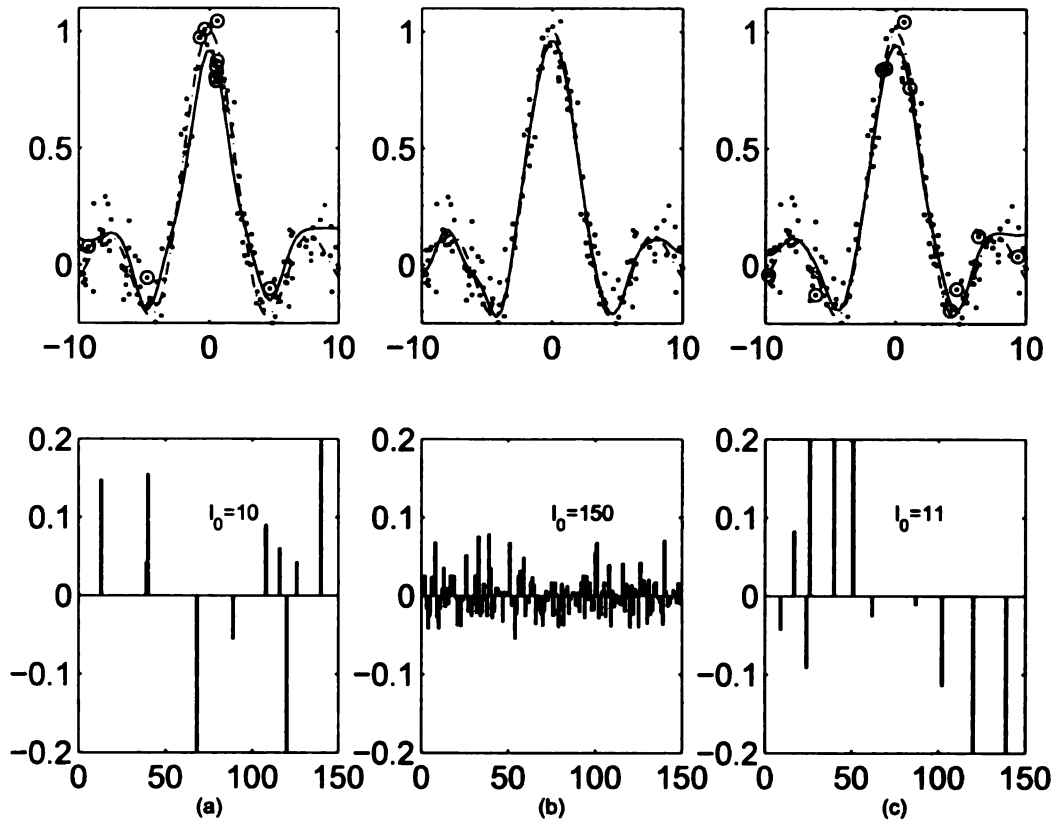


Figure 4.2: Kernel regression results. The dashed lines are true sinc functions. Solid lines are approximation results. The circles shown in the first row are those selected kernel functions. Note that we do not show the circles in the ridge regression results, because every kernel is selected. (a) Proposed gradient method. (b) Ridge regression. (c) LASSO regression.

from 0.3 to 1.5. As indicated in the figure, greater t makes the model fit well but increases its complexity, and vice versa.

4.3.2 Classification

The experiments addressed the kernel-based classifier for two-class problems: A special case of the regression problem with $y \in \{+, -\}$. The classifier is formulated as

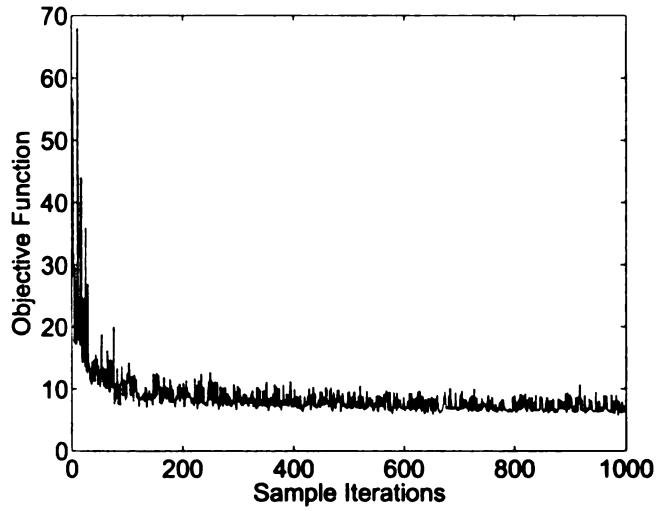


Figure 4.3: Objective function vs. number of iterations. We utilize a dynamic learning rate mechanism called Amnesic Average. Interested readers can refer to (Weng and Zhang, 2004) for more details.

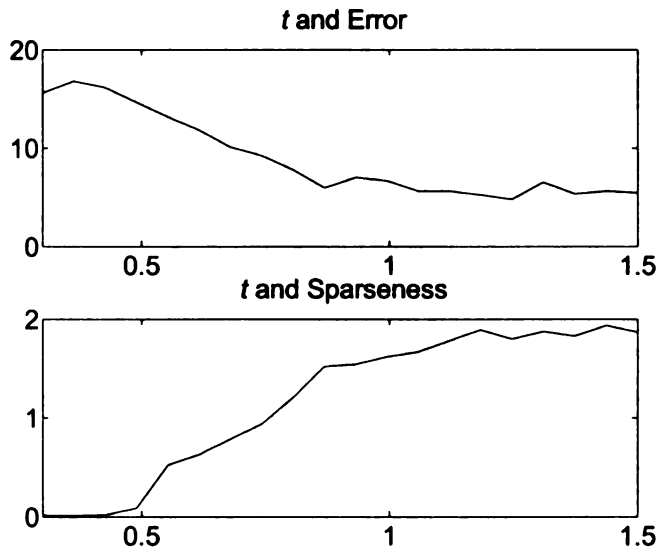


Figure 4.4: The effect of control parameter t over mean square error and sparseness.

the following regression function:

$$p(y \mid \mathbf{x}) = \psi(\beta_0 + \sum_{i=1}^{k} \beta_i K(\mathbf{x}, \mathbf{x}_i))$$

where ψ denotes the logistic function $\psi(z) = (1 + \exp(-z))^{-1}$. If $p(y \mid \mathbf{x}) > 0.5$, $\mathbf{x}$ belongs to the class $+$. Otherwise, $\mathbf{x}$ belongs to the class $-$.

First, we run the classifier on an artificial data set drawn randomly from the 2D XOR function: $y = \text{sign}(x_1 * x_2)$, where $x_1, x_2 \in (-1, 1)$. The data set includes 30 samples.

Two methods are compared. In Fig. 4.5 (a), the top row shows the decision boundary of our proposed method while the bottom one illustrates the values of the learned coefficient vector β. For comparison, (b) gives the results using the LSSVM (Suykens and Vandewalle, 1999). The sparseness is improved tremendously.

Second, we use three data sets from real-data problems: the *Pima indian diabetes*[1], which are collected from women of Pima heritage and the goal is to decide whether a subject has diabetes or not, based on 7 different tests; the *Wisconsin breast cancer* (WBC)[2], whose goal is to diagnose (benign/malignant) based on the results of 9 measurements; the AR face database[3], in which two male face samples were selected and each person has 26 samples. In WBC, we remove the cases with missing attributes for simplicity. Table 4.3.2 shows the results of the proposed classifier on the 5-fold cross validation experiment. For comparison, we also include the results of the

[1]Downloadable at www.stats.ox.ac.uk/pub/PRNN
[2]Downloadable at www.ics.uci.edu/ mlearn/MLSummary.html
[3]Available at rvl1.ecn.purdue.edu/v1/ARdatabase/ARdatabase.html

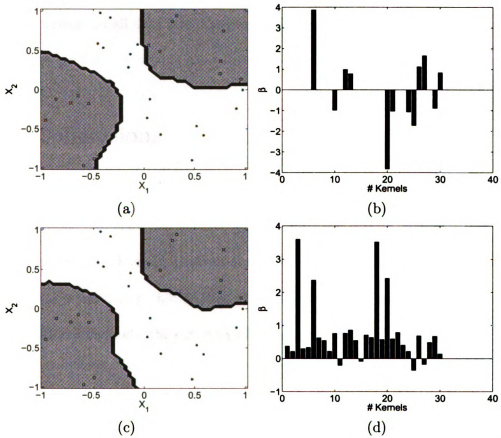


Figure 4.5: The decision boundary of the artificial XOR data set. (a) The decision boundary of the proposed winner-take-all method. (b) Kernel selection of the proposed method. (c) The decision boundary of Least Square Support Vector Machine (LSSVM). (d) Kernel selection of LSSVM.

Table 4.1: The result of the 5-fold cross-validation.

ROC/No. kernels	Pima	WBC	DBF face
Logistic	0.75/200	0.77/455	0.81/40
LSSVM	0.90/200	0.76/455	0.96/40
Proposed classifier	0.96/70	0.98/253	0.97/20

logistic classifier. In all data sets, the proposed classifier is better. The performance is improved partially by setting some decayed kernel weights to be exactly zero.

4.4 Conclusions

In this chapter, we have formulated the dual problem of the ℓ_1 norm based sparse approximation. We show the geometric relation of the duality and solve the dual ℓ_∞ maximization problem by gradient and quadratic programming. The algorithm's performance is close to or better than the result of LASSO regression. Compared with traditional methods, the efficiency of this algorithm is quite remarkable.

Chapter 5

Conclusions

5.1 Summary

This dissertation presents a general-purpose vision architecture shown in Fig. 5.1, which can accommodates attention selection and development of a vision system. Motivated by the structure of the early visual pathway, several layers of staggered receptive fields are used to model the pattern of positioning of the processing cells of the early visual pathway. From sequentially arriving video frames the proposed SHM algorithm develops a hierarchy of filter banks, whose outputs are uncorrelated within each layer, but with an increasing scale of receptive fields that range from low to high layers. We also show how this general architecture can be applied to occluded face recognition, which demonstrates a case of attention selection.

The architecture shown in Fig. 5.1 is equally well applicable to other sensory modalities, such as vision, speech and touch, each with a different set of developmental parameters (e.g., the extent of temporal context).

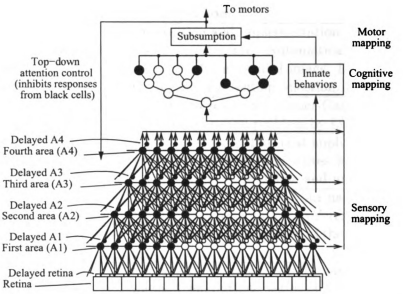


Figure 5.1: The flow diagram of a developmental vision system. The sensorimotor module for innate behavior is illustrated by a block only for simplicity, but it also contains the three mappings which are, however, much smaller in storage and much simpler in the mappings it realizes.

Besides the general architecture, a developmental algorithm for sensory network has been designed and tested based on the concept of lobe components. Our simulation result of network development revealed that many *lobe component cells* developed by the algorithm from natural images are orientational selective cells similar to those found in V1.

The proposed LCA method is based on two well-known computational neural mechanisms, Hebbian learning and lateral inhibition. It is an in-place developmental algorithm in which every cell is both a signal processor and the developer of the cortex. There is no need for a separate network to deal with the development (and learning). The algorithm is purely incremental in the sense that there is no need for extra storage for information other than that can be stored and incrementally computed by the processing network itself.

Topic	Contribution
General vision architecture	Proposition and implementation of a general vision architecture; Automatically developing filters from natural image; Demonstrating a case of attention selection.
Blind source separation	An efficient solution to Super-Gaussian Blind Source Separation problem by LCA method.
Feature derivation	A fast LCA algorithms that applies to spatial filter development, and solves local feature extraction problem online and in realtime.
Natural image filters	Developing LCA filters from natural image set; Showing orientation preference neurons and the pinwheel structure of the filter topology.
Feature Selection	Sparse optimization and winner-take-all method for optimal selection of basis functions and features.

Table 5.1: Contribution of the dissertation.

we further discuss the sparseness optimization. We propose a new perspective to achieve sparseness via the winner-take-all principle for the linear kernel regression and classification task. We form the duality of the LASSO criteria, and convert an ℓ_1 norm minimization to an ℓ_∞ norm maximization problem. The winner-take-all updating rule coming from the ℓ_∞ term is especially efficient due the simplicity of the maximum function. This method has been successfully tested on several tasks, including base/feature selection, classification and regression.

5.2 Contributions

The major contribution of this dissertation can be summarized in Table 5.1

5.3 Summary of the future work

The research work done in this dissertation shows that a general purpose vision system is possible to solve a variety of vision problems. However, some issues need to be explored further:

1. An interesting research direction in ICA is independent subspace analysis, in which the components are divided into groups or subspaces so that components in different subspaces are independent, but components in the same subspace are not. In particular, the distribution of the components in a subspace is assumed to depend only on the norm of the projection on that subspace. Typically, this implies that the components of a subspace tend to be active simultaneously. The proposed LCA algorithm could be applied to this problem. In the original LCA algorithm, only one winner will be updated to "take care of" the sample in the space, therefore leads to the direction of the maximum sparseness. If we modify the algorithm to update the top n winners among all the vectors that has the maximum subspace projection norm then we could find the sparseness structure in this subspace representation. At current stage we have not yet test this hypothesis, whereas it is a reasonable research direction for the future work.

2. Other basis development algorithm should also be considered and explored, including but not limit to Linear Discriminant Analysis (LDA), non-linear PCA and etc.

3. Temporal information is extremely important in perception, however in the

current model we don't deal with much temporal information. A promising research direction is to adopt the temporal information and look for spatio-temporal filter development algorithm that can catch the structure in both spatial and temporal domain.

4. A general purpose vision system including an attention selection mechanism that accommodates both top-down and bottom-up attention selection mechanism should be tested in real environment. This architecture will serve as the sensory mapping part of the autonomous mental development (AMD) paradigm.

Appendix A

The Candid Covariance-free Incremental PCA Algorithm

The main goal of the CCIPCA (Weng et al., 2003) algorithm is to compute the principal component vectors incrementally without estimating the covariance matrix of input vectors. Similar techniques can be dated back to (Oja, 1982) and (Sanger, 1989). However, in this method, the statistically efficient estimation is considered. Thus, it is much faster than the previous learning-rate based method. The following algorithm does not assume staggering, but staggering only requires that the residual images are tiled together before the next principal vector is computed at a shifted position.

In this algorithm, the principal components of the input are computed incrementally. Each input is a column vector $x_1, x_2, ..., x_n, ...$, with dimension d. The covariance matrix A of the vectors is unknown and cannot be estimated. An incremental approximation is needed of the first k largest eigenvalues $\sigma^{(n)} = (\lambda_1^{(n)}, \lambda_2^{(n)}, ..., \lambda_k^{(n)})$

and the associated eigenvectors $V^{(n)} = [v_1^{(n)}, v_2^{(n)}, ..., v_k^{(n)}]$, when $n \geq k$, where each $v_i^{(n)}$ is a d-dimensional column vector, thus $V^{(n)}$ has k columns. Since each $x_1, x_2, ..., x_n$ would require too much space to retain, all operations are done incrementally. The previous eigenvalues $\sigma^{(n-1)}$ and previous eigenvectors $V^{(n-1)}$ must be updated by only using the new sample input x_n.

CCIPCA algorithm

1: Choose the initial guess of $\bar{x}^{(0)}$ and k, where $\bar{x}^{(0)}$ is the samples mean and k is number of eigenvectors with the largest k eigenvalues.

2: Choose the initial guess of $V^{(0)}$. For example, it can be any normalized random image. Choose the initial guess $\lambda_i^{(0)} = 0, i = 1, 2, ...k$.

3: **for** $n = 1, 2, 3, ...N$, where N is total number of samples **do**

4: Get a new sample x_n

5: Use the following incremental method to compute the amnesic average $\bar{x}^{(n)}$:

$$\bar{x}^{(n)} = \frac{n-1-l}{n}\bar{x}^{(n-1)} + \frac{1+l}{n}x_n \tag{A.1}$$

{where l is the amnesic average parameter. When $n-1-l \leq 0$ use non-amnesic averaging (the above equation with l set temporarily to zero).}

6: Compute the scatter vector: $u_n = x_n - \bar{x}^{(n)}$

7: Set the first residual vector $\hat{u}_1 = u_n$

{Now we proceed to update the ith principal vectors and projecting the residual vector. Between each iteration, the residual vector (next receptive field) is returned to a large residual image before the next filter is extracted. At this

point, shifting can occur to implement the "scattered" receptive field method.}

8: **for** $i = 1, 2, 3..., k$ **do**

9: Update the ith principal vector:

$$V_i^{(n)} = \alpha(n) V_i^{(n-1)} + \beta(n) \frac{1}{\left\| V_i^{(n-1)} \right\|} \left(\hat{u}_i \cdot V_i^{(n-1)} \right) \hat{u}_i \quad , \tag{A.2}$$

where

$$\begin{aligned} \alpha(n) &= \tfrac{n-1-l}{n} \\ \beta(n) &= \tfrac{1+l}{n} \end{aligned} \tag{A.3}$$

are the coefficient of the amnesic average, and l is a constant.

10: Compute the coordinate of the residual vector along the current ith principal component:

$$y_i = \frac{1}{\| V_i^{(n)} \|} (\hat{u}_i \cdot V_i^{(n)}) \tag{A.4}$$

11: Update the ith eigenvalue:

$$\lambda_i^{(n)} = \alpha(n) \lambda_i^{(n-1)} + \beta(n) y_i^2 \tag{A.5}$$

12: Compute the remaining residual vector (next input):

$$\hat{u}_{i+1} = \hat{u}_i - \frac{y_i \cdot V_i^{(n)}}{\| V_i \|} \tag{A.6}$$

13: **end for**

14: **end for**

The eigenvectors that are incrementally developed by this algorithm represent the basis of the sample space. Their respective eigenvalues represent the weight of each of the filters and are used to normalize them so that each filter has the same weight when computing a response to the input. This procedure is also referred to as "whitening." After whitening, the output of each filter will be uncorrelated and has an identity covariance matrix.

The whitened filter output is computed as follows:

$$\tilde{y}_i = \frac{y_i}{\sqrt{\lambda_i^{(n)}}}$$

Appendix B

Derivative of Maximum Function

To simplify the derivation, let

$$\begin{cases} f_i(\beta) = \left(\frac{2}{t}(\beta^T \mathbf{C}_i - \mathbf{y}^T \mathbf{H}_i)\right)^2, i < n \\ f_i(\beta) = 1, i = n \end{cases} \tag{B.1}$$

Then

$$\max_i \{f_i(\beta)\} = \lim_{r \to \infty} \left[\sum_i f_i(\beta)^r\right]^{\frac{1}{r}}.$$

The derivative of the maximum function can be defined by,

$$\begin{aligned}
&\frac{\partial}{\partial\beta}\lim_{r\to\infty}\left[\sum_i f_i(\beta)^r\right]^{\frac{1}{r}} \\
&= \lim_{r\to\infty}\left\{\left[\sum_i f_i(\beta)^r\right]^{\frac{1}{r}-1}\cdot\sum_{i=1} f_i(\beta)^{r-1}\frac{\partial f_i(\beta)}{\partial\beta}\right\} \\
&= \lim_{r\to\infty}\left\{\left[\sum_i f_i(\beta)^r\right]^{\frac{1}{r}}\cdot\frac{\sum_i f_i(\beta)^{r-1}\frac{\partial f_i(\beta)}{\partial\beta}}{\sum_i f_i(\beta)^r}\right\}
\end{aligned} \tag{B.2}$$

Let

$$A = \frac{\sum_i f_i(\beta)^{r-1}\frac{\partial f_i(\beta)}{\partial\beta}}{\sum_i f_i(\beta)^r}, \tag{B.3}$$

and let $m = \arg\max_j \{f_i(\beta)\}$, then

$$A = \frac{\sum_i \frac{f_i(\beta)^{r-1}}{f_m(\beta)^{r-1}}\frac{\partial f_i(\beta)}{\partial\beta}}{\sum_i \frac{f_i(\beta)^r}{f_m(\beta)^{r-1}}} \tag{B.4}$$

Notice that when $r\to\infty$, only the term $f_m(\beta)^{r-1}/f_m(\beta)^{r-1}$ remains, and all other terms vanishes. So, we have

$$\lim_{r\to\infty} A = \frac{1}{f_m(\beta)}\cdot\frac{\partial f_m(\beta)}{\partial\beta}.$$

We obtain the derivative of maximum function

$$\frac{\partial}{\partial\beta}\max\{f_i(\beta)\} = \frac{\partial f_m(\beta)}{\partial\beta}, \tag{B.5}$$

where $m = \arg\max_j \{f_i(\beta)\}$.

Bibliography

Adrian, E. D. (1941). Afferent discharges to the cerebral cortex from peripheral sense organs. *J. of Physiology*, 100:154–191.

Aloimonos, Y. (1992). Purposive active vision. *Computer Vision, Graphics, and Image Processing: Image Understanding*, 17(1-3):285–348.

Amaldi, E. and Kann, V. (1998). On the approximibility of minimizing non zero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209:237–260.

Amari, S. (1977). Dynamics of pattern formation in lateral-inhibition type neural fields. *Biological Cybernetics*, 27:77–87.

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276.

Amari, S.-I., Cichocki, A., and Yang, H. (1996). A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems 8*, pages 757–763. MIT Press.

Atick, J. J. and Redlich, A. N. (1993). Convergent algorithm for sensory receptive field development. *Neural Computation*, 5:45–60.

Backer, G., Mertsching, B., and Bollmann, M. (2001). Data- and model-driven gaze control for an active-vision system. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(12):1415–1429.

Barlow, H. B. (1961). Possible principles underlying the transformations of sensory messages. In Rosenblith, W. A., editor, *Sensory Communication*, pages 217–234. MIT Press.

Barlow, H. B. (1972). Single units and sensation: a neurone doctrine for perceptual psychology? *Perception*, 1:371–394.

Bartsch, A. P. and van Hemmen, J. L. (2001). Combined hebbian development of geniculocortical and lateral connectivity in a model of primary visual cortex. *Biological Cybernetics*, 84:41–55.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

Bell, A. J. and Sejnowski, T. J. (1997). The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford Unversity Press, Oxford.

Blakemore, C. and Cooper, G. F. (1970). Development of the brain depends on the visual environment. *Nature*, 228:477–478.

Blaschke, T. and Wiskott, L. (2002). An improved cumulant based method for independent component analysis. In Dorronsoro, J. R., editor, *Proc. Intl. Conf. on Artificial Neural Networks - ICANN'02*, Lecture Notes in Computer Science, pages 1087–1093. Springer.

Burger, T. and Lang, E. (1999). An incremental Hebbian learning model of the primary visual cortex with lateral plasticity and real input patterns. *Zeitschrift fur Naturforschung C-A Journal of Biosciences*, 54:128–140.

Cardoso, J.-F. (1997). Infomax and maximum likelihood for source separation. *IEEE Signal Processing Letters*, 4:112–114.

Cardoso, J.-F. and Laheld, B. H. (1996). Equivariant adaptive source separation. *IEEE Trans. on Signal Processing*, 44(12):3017–3030.

Christianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press, Cambridge, UK.

Comon, P. (1994). Independent component analysis, A new concept? *Signal Processing*, 36:287–314.

Crair, M., Gillespie, D., and Stryker, M. (1998). The role of visual experience in the development of columns in cat visual cortex. *Science*, 279:566–570.

Desimone, R. and Duncan, J. (1995). Neural mechanism of selective visual attention. *Annual Review Neuroscience*, 18:193–222.

Ditterich, J., Eggert, T., and Straube, A. (2000). The rule of attention focus in the visual information processing underlying saccadic adaptation. *Vision Research*, 40:1125–1134.

Elman, J., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1997). *Rethinking Innateness: A connectionist perspective on development.* MIT Press, Cambridge, Massachusetts.

Eriksen, C. and Yeh, Y. (1985). Allocation of attention in the visual field. *Journal of experimental Psychology: Human Perception and Performance*, 11(5):583–597.

Field, D. J. (1994). What is goal of sensory coding? *Neural Computation*, 6:559–601.

Figueiredo, M. (2003). Adaptive sparseness for supervised learning. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(9):1150–1159.

Friedman, J. (1987). Exploratory projection pursuit. *J. of the American Statistical Association*, 82(397):249–266.

Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. of Computers*, c-23(9):881–890.

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202.

Gersho, A. (1979). Asymptotically optimal block guantization. *IEEE Trans. on Information Theory*, 25(4):373–380.

Giannakopoulos, X., Karhunen, J., and Oja, E. (1998). Experimental comparison of neural ICA algorithms. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*, pages 651–656, Skvde, Sweden.

Gray, R. M. and Neuhoff, D. L. (1998). Guantization. *IEEE Trans. on Information Theory*, 44(6):1–63.

Hebb, D. O. (1949). *The organization of behavior; a neuropsychological theory.* Wiley, New York.

Herault, J. and Jutten, J. (1986). Space or time adaptive signal processing by neural network models. In Denker, J. S., editor, *Neural Networks for Computing: AIP Conference Proceedings*, volume 151, New York. American Institute for Physics.

Hoerl, A. and Kennard, R. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67.

Hopfinger, J., Buonocore, M., and Mangun, G. (2000). The neural mechanism of top-down attention control. *Nature Neuroscience*, 3(3):284–291.

Hubel, D. and Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. of Physiology*, 160:106–154.

Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics*, 13(2):435–475.

Hwang, W. S. and Weng, J. (2000). Hierarchical discriminant regression. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1277–1293.

Hyvärinen, A. (1998). New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems*, volume 10, pages 273–279. MIT Press.

Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128.

Hyvärinen, A. (2001). Fast independent component analysis. In Roberts, S. and Everson, R., editors, *Independent Component Analysis: Principles and Practice.* Cambridge University Press. in press.

Hyvärinen, A. and Hoyer, P. O. (2000). Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720.

Hyvärinen, A. and Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492.

Hyvärinen, A. and Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural Networks*, 13(4-5):411–430.

Itti, L., Gold, C., and Koch, C. (2001). Visual attention and target detection in clutterd natural scenes. *Optical Engineering*, 40(9):1784–1793.

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Itti, L., Koch, C., and Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(11):1254–1259.

James, W. (1890/1981). *The Principles of Psychology.* Harvard University Press, Cambridge, MA.

Jolliffe, I. T. (1986). *Principal Component Analysis.* Springer-Verlag, New York.

Jones, M. and Sibson, R. (1987). What is projection pursuit ? *J. of the Royal Statistical Society, Ser. A*, 150:1–36.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M., editors (2000). *Principles of Neural Science.* McGraw-Hill, New York, 4th edition.

Karhunen, J. and Pajunen, P. (1997). Blind source separation using least-squares type adaptive algorithms. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pages 3048–3051, Munich, Germany.

Karhunen, J., Pajunen, P., and Oja, E. (1998). The nonlinear pca criterion in blind source separation: Relations with other approaches. *Neurocomputing*, 22:5–20.

Karvanen, J. and Cichocki, A. (2003). Measuring sparseness of noisy signals. In *Proceedings of 4th International symposium on Independent Component Analysis and Blind Signal Seperation (ICA2003)*, pages 125–130.

Kass, J. (1997). Topographic maps are fundamental to sensory processing. *Brain Research Bulletin*, 44(2):107–112.

Kirby, M. and Sirovich, L. (1990). Application of the karhunen-loève procedure for the characterization of human faces. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(1):103–108.

Koch, C. and Ullman, S. (1985). Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neuronbiology*, 4:219–227.

Kohonen, T. (2001). *Self-organizing Maps*. Springer-Verlag, New York. 3rd edition.

Kullback, S. (1959). *Information Theory and Statistics*. Wiley.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

Lee, T. W., Girolami, M., and Sejnowski, T. J. (1999). Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation*, 11(2):417–441.

Lehmann, E. L. (1983). *Theory of Point Estimation*. John Wiley and Sons, Inc., New York.

Lin, Y. (2002). Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6:259–275.

Linsker, R. (1986). From basic network principles to neural architecture: emergence of spatial-oponent cells. *Proc. Natl. Acad. Sci.*, 83:7508–7512, 8390–8394, 8779–8783.

Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, pages 105–117.

Linsker, R. (1992). Local synaptic learning rules suffice to maximize mutual information in a linear network. *Neural Computation*, 4:691–702.

Luenverger, D. G. (1969). *Optimization by Vector Space Methods*. John Wiley and Sons, Inc.

Marshall, W. and Talbot, S. (1942). Recent evidence for neural mechanisms in vision leading to a general theory of sensory acuity. *Bio. Symp.*, 7:117–164.

McLachlan, G. J. (1997). *The EM Algorithm and Extensions*. Wiley, New York.

Miller, K. D. (1994). A model for the development of simple cell receptive fields and the odered arrangement of orientation columns through activity-dependent competition between on- and off-center inputs. *J. of Neuroscience*, 14:409–441.

Nadal, J.-P. and Parga, N. (1994). Non-linear neurons in the low noise limit: a factorial code maximizes information transfer. *Network*, 5:565–581.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *J. of Mathematical Biology*, 15:267–273.

Olshausen, B., Anderson, C., and Essen, D. V. (1993). A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *Journal of Neuroscience*, 13(11):4700–4719.

Olshaushen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609.

Olshaushen, B. A. and Field, D. J. (1997). Sparse coding with an overcomplete basis set: A strategy used by v1? *Vision Research*, 37(23):3311–3325.

Pajunen, P. and Karhunen, J. (1998). Least-squares methods for blind source separation based on nonlinear PCA. *Int. J. of Neural Systems*, 8(5-6):601–612.

Papoulis, A. (1991). *Probability, random variables, and stochastic processes.* McGraw-Hill, New York. 3rd edition.

Pashler, H., editor (1996). *Attention.* University College London, London.

Pearlmutter, B. A. and Parra, L. C. (1997). Maximum likelihood blind source separation: a context-sensitive generalization of ICA. In *Advances in Neural Information Processing Systems*, volume 9, pages 613–619.

Pham, D.-T. and Garrat, P. (1997). Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. on Signal Processing*, 45(7):1712–1725.

Pham, D.-T., Garrat, P., and Jutten, C. (1992). Separation of a mixture of independent sources through a maximum likelihood approach. In *Proc. EUSIPCO*, pages 771–774.

Roth, Z. and Baram, Y. (1996). Multidimensional density shaping by sigmoids. *IEEE Trans. on Neural Networks*, 7(5):1291–1298.

Rubner, J. and Schulten, K. (1990). Development of feature detectors by self-organization. *Biological Cybernetics*, 62:193–199.

Sanger, T. D. (1989). Optimal unsupervised learning in a single-layer linear feedforward neural network. *IEEE Trans. on Neural Networks*, 2(7):459–473.

Schill, K., Umkehrer, E., Beinlich, S., Krieger, G., and Zetzsche, C. (2001). Scene analysis with saccadic eye movemens: Top-down and bottom-up modeling. *Journal of Electronic Imaging*, 10(1):152–160.

Sillito, A., Grieve, K., Jones, H., Cudeiro, J., and Davis, J. (1995). Visual cortical mechanisms detecting focal orientation discontinuities. *Nature*, 378:492–496.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

Sirosh, J. and Miikkulainen, R. (1997). Topographic receptive field and patterned lateral interaction in a self-organizing model of the primary visual cortex. *Neural Computation*, 9:577–594.

Sur, M., Angelucci, A., and Sharm, J. (1999). Rewiring cortex: The role of patterned activity in development and plasticity of neocortical circuits. *Journal of Neurobiology*, 41:33–43.

Suykens, J. A. K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Journal of Neural Processing Letters*, 9(3):293–300.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *J. Royal Statistical Society (B)*, 58:267–288.

Tipping, M. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244.

Treisman, A. (1986). Features and objects in visual processing. *Scientific American*, 255(5):114–125.

Treisman, A. and Gelade, G. (1980). A feature-intergration theory of attention. *Cognitive Pshchology*, 12(1):97–136.

Tsotsos, J., Culhane, S., Wai, W., Lai, Y., Davis, N., and Nuflo, F. (1995). Modeling visual attention via selective tuning. *Artificial Intelligence*, 78:507–545.

Tucker, H. G. (1962). *An introduction to probability and mathematical statistics.* Academic Press, New York.

Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86.

Ungerleider, L. and Mishkin, M. (1980). Two cortical visual systems. In Ingle, D. J., Goodale, M. A., and Mansfield, R. J. W., editors, *Analysis of Visual Behavior.* MIT Press, Cambridge, MA.

von der Malsburg, C. (1973). Self-organization of orientation-sensitive cells in the striate cortex. *Kybernetik*, 15:85–100.

Weinberg, R. J. (1997). Are topographic maps fundamental to sensory processing? *Brain Rearch Bulletin*, 44(2):113–116.

Weng, J., Ahuja, N., and Huang, T. (1997). Learning recognition and segmentation using the cresceptron. *International Journal of Computer Vision*, 25(2):109–143.

Weng, J., Huang, T. S., and Ahuja, N. (1993). *Motion and Structure from Image Sequences.* Springer-Verlag, New York.

Weng, J. and Hwang, W. (2002). Online image classification using IHDR. *International Journal on Document Analysis and Recognition*, 5(2-3):118–125.

Weng, J., McClelland, J., Pentland, A., Sporns, O., Stockman, I., Sur, M., and Thelen, E. (2001). Autonomous mental development by robots and animals. *Science*, 291(5504):599–600.

Weng, J. and Stockman, I. (2002). Autonomous mental development: Workshop on development and learning. *AI Magazine*, 23(2):95–98.

Weng, J. and Zhang, N. (2004). A quasi-optimally efficient algorithm for independent component analysis. In *Proc. IEEE Int. Conf. on Acoustics Speech, and Signal Processing*, Montreal, Canada.

Weng, J., Zhang, Y., and Hwang, W. (2003). Candid covariance-free incremental principal component analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(8):1034–1040.

Wiskott, L. and Sejnowski, T. (2002). Slow feature analysis: unsupervised learning of invariances. *Neural Computation*, 14(4):715–770.

Zador, P. L. (1982). Asymptotic quantization error of continuous signals and the quantization dimension. *IEEE Trans. on Information Theory*, 28(2):139–149.

Zhang, N. and Weng, J. (2004). Sparse representation from a winner-take-all neural network. In *Proc. IEEE Int. Joint Conf. on Neural Networks*, Budapest, Hungary.