STRUCTURE AND SIMILARITY IN VISION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Reza Shahbazi

Jan 2015

STRUCTURE AND SIMILARITY IN VISION

Reza Shahbazi, Ph.D.
Cornell University 2015

A proper account of the neural computations involved in visual cognition must address, among other things, structure and similarity. In this work we present a set of behavioral experiments that investigate the effect of hierarchic structures in visual cognition, and suggest that learning the correct category of visual stimuli is easier when their underlying generative distributions are organized more hierarchically. We then go on to discuss the implications of the Reproducing Kernel Hilbert Space theory for measurements of similarity in neural settings, and discuss a number of empirical findings that suggest the brain in fact relies on kernel-like computations at various stages of processing.

BIOGRAPHICAL SKETCH

Reza Shahbazi studied Cognitive Science at the University of California in San Diego, with a specialization in computation, and received his bachelor's degree in May 2008. In August 2008 he started his graduate studies at Cornell University in the field of psychology under Shimon Edelman's supervision.

This document is dedicated to every distal and proximal factor involved in its getting to this point, and Bach.

ACKNOWLEDGEMENTS

This work would not have been possible without the generosity of Cornell University, and the department of Psychology. I am particularly grateful to David Field for his invaluable advice over the years, and James Cutting, Ashutosh Saxena, Michael Nussbaum, and Tom Gilovich, and the undergraduate research assistants that helped me with running the experiments, and to the graduate students of Psychology.

Nothing in here would have concluded without the infinite wisdom, patience, and kindness of Shimon Edelman.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

More than thirty years after Marr's ambitious *Vision* (Marr, 1982), compiling a systematic theory of the principles of visual cognition still seems like an ambitious undertaking. What is missing is not faster or more parallel computing, nor sophistication in method. The growing popularity of deep architectures of learning, coupled with the massively parallel computing clusters–sometimes spanning 16000 cores (Le, 2013)–make possible machinery that exhibits respectable performance in various domains of object detection and identification. In parallel, computational techniques have reached such sophistication that it takes graduate-level training in mathematics to fully understand them–e.g., Dirichlet Process Mixture Modeling, (Neal, 2000). These advancements have led to markedly improved artificial systems, some good enough to be integrated into consumer products such as digital cameras. Yet in spite of their merits, our current algorithms fall short compared to human vision. For instance, with the exception of a few particular domains such as detection of faces, cars, and pedestrians, object recognition is still unreliable, especially when applied to natural images, where problems such as clutter and variable pose abound. Furthermore, to reach their optimum performance, existing systems require massive training. In contrast, human vision exhibits robust performance, and appears to need few examples to do so.

The roots of this discrepancy can be traced, in part, to the minor role that *structure* plays in today's artificial vision systems. The state of the art approaches in computer vision tend to focus on "bag of features" methods–e.g., SIFT, (Lowe, 1999), and various forms of mixture modeling–whose treatment of structural information is either implicit or altogether absent (Poggio and Ullman, 2013). This is in contrast to human vision's aptitude to exploit structural clues to aid recognition, especially when dealing with composite stimuli such as scenes or novel objects made of familiar parts. Indeed, converging evidence from anatomical, functional, and behavioral experiments suggest that one particular type of structure, namely, hierarchy, plays an important role in visual cognition.

In the feedforward cortical visual stream of mammalian brain, receptive fields at each successive stage are formed by gathering input from multiple units at preceding stages, resulting in an organization that resembles a hierarchy both in anatomy and function. Does our hierarchically organized visual system favor hierarchically structured stimuli?

The answer, according to a set of behavioral experiments we have conducted, is in the affirmative. Chapters 2 and 3 report a set of experiments that were devised to investigate the interaction of hierarchic structures with visual learning. Chapter 2 sets forth the general paradigm of the experiments, and covers some preliminary data suggesting that hierarchically organized visual stimuli can be learned more accurately than their non-hierarchic counterparts, in a classification task. In chapter 3 we introduce a graph theoretic measure of hierarchicality to explore the gradation of the effect observed earlier. Furthermore, in order to more confidently attribute the observations to the structural differences of the experimental conditions, we employ a number of techniques to equalize the difficulty induced by non-structural factors.

Although machine vision may not be quite up to its human counterpart in respecting structure, when it comes to similarity-based representations, it is the natural vision researches that can benefit by following in the footsteps of artificial systems. For at least as far back as John Locke's *Essay on Human Understanding* (Locke, 1700) followed later by Hume, and more recently by Shepard, similarity has been known to be play a frontal role in what the mind does and how it does it. Yet rigorous treatment of similarity has not been as pronounced in brain sciences as in machine learning, where it takes the form of *kernel* methods discussed in the theory of Reproducing Kernel Hilbert Space.

This idea is explored in chapter 4. We introduce four fundamental constraints on cognition, and discuss how regarding kernels as a measure of similarity (as opposed to their more popular view as a frugal trick in high dimensional computations) can simultaneously meet the requirements of all four constraints.

CHAPTER 2

THE ROLE OF HIERARCHY IN LEARNING TO CATEGORIZE IMAGES

Abstract Converging evidence from anatomical studies (Maunsell and van Essen, 1983) and functional analyses (Hubel and Wiesel, 1968) of the nervous system suggests that the feed-forward pathway of the mammalian perceptual system follows a largely hierarchic organization scheme. This may be because hierarchic structures are intrinsically more viable and thus more likely to evolve (Simon, 1973, 2002). But it may also be because objects in our environment have a hierarchic structure and the perceptual system has evolved to match it. We conducted a behavioral experiment to investigate the effect of the degree of hierarchy of the generative probabilistic structure in categorization. We generated one set of stimuli using a hierarchic underlying probability distribution, and another set according to a non-hierarchic one. Participants were instructed to categorize these images into one of the two possible categories a. Our results suggest that participants perform more accurately in the case of hierarchically structured stimuli[1].

2.1 Regarding Hierarchies

The anatomy of the primate visual system suggests that the retinal input progresses through several stages of processing that form an approximate hierarchy. In the visual system, a large number of photoreceptors project to one ganglion cell, several of which converge onto a single LGN cell; then come the cortical areas V1, V2, IT, etc. (Kaiser and Hilgetag, 2010; Modha and Singh, 2010).

[1] This chapter is based on an article co-authored with David Field and Shimon Edelman (Shahbazi, Field, and Edelman, 2011).

The impression of hierarchy is further strengthened by evidence from functional analysis of the neuronal circuits. For instance, in V1 several simple cells send their axons to one complex cell whose preferred stimulus is constructed by the preferred stimuli of its input simple cells (Hubel and Wiesel, 1968). Moreover, starting from the retina and going up to higher cortical areas, the complexity of the features that each stage of this hierarchy responds best to increases (Gross, Rocha-Miranda, and Bender, 1972).

There exist at least three different definitions of hierarchy in the literature. According to the most parsimonious of them, a hierarchy is any system of items where no item is superior to itself. Furthermore, there needs to be one hierarch, an item which is superior to all other items (Dawkins, 1976). This definition emphasizes that aspect of hierarchy that differentiates it from a heterarchy (McCulloch, 1945). According to McCulloch, heterarchy is a structure with a certain circularity. This circularity results in the possibility of members of the system being superior to themselves. Because of the paradoxes that it may engender, heterarchy is an unlikely structure to be observed in our everyday lives, hence the name (heterarchy is Greek for “under the governance of an alien”; von Goldammer, Paul, and Newbury, 2003). Another definition of hierarchy comes from algebra, where hierarchies are defined in terms of partially ordered sets (posets; Lehmann, 1996). The third definition is the one advocated by Herbert Simon (1973), the pioneering figure of hierarchy theory. While the three definitions are not in disagreement with each other, the third one seems to be best suited for the present discussion.

According to Simon, a hierarchy is a nested collection of items where each item contains another set of subcollections. He uses the analogy of Chinese boxes, in which each box contains several smaller boxes while it is itself contained, together with other boxes, in a larger one. Graphically, this resembles the structure of a tree where vertices represent items and edges indicate containment. At least since the mid twentieth century, hierarchies have been believed to be the appropriate structure for the organization of complex systems in various domains including sociology, biology, computer science, and cognitive science (Simon, 1973; Hirtle and Jonides, 1985; Holling, 2001).

In cognitive science, neuroanatomical data are one source of the evidence for the hierarchic structure of the visual system. Another line of evidence come from computational considerations. The problem of inferring the state of the environment from the sensory input is an ill posed problem (Chater, Tenenbaum, and Yuille, 2006; Edelman, 2008). The normative approach to this problem is to rely on the environmental statistics that have been acquired via past experience. A cognitive system that relies on the statistics of its environment to perform its tasks will soon run out of resources as the computational cost of keeping the joint statistics of the environmental variables grows exponentially in the number of variables that the system is keeping track of (an issue known as the curse of dimensionality; Bishop, 2006). By employing a hierarchic structure in recording the statistics, the system can bring the computational cost of the task under control. In addition to this computational advantage, hierarchic systems have been shown to be more stable and evolve faster than their alternatives (Simon, 2002).

While on the one hand it is inherently beneficial for systems to have a hierarchic structure, on the other hand, specifically in the case of perception, it is beneficial for a system to employ a hierarchic structure to represent its environment. Indeed, in the visual domain objects seem to present themselves to us in a hierarchic way. For example, a face is composed of two eyes, one nose, one mouth etc.; an eye is in turn composed of the iris, pupil, eyelashes etc. Is this hierarchy merely apparent, simply because of the hierarchic structure of our own perceptual system, or is it truly "out there"?.

In this paper we address this question indirectly, by evaluating the effect of the interaction between the probabilistic hierarchic structure that we build into a family of stimuli and the ability of human subjects to categorize those stimuli. In a series of related studies Orban and colleagues have investigated learning of visual scenes in human subjects where higher level features are formed, in a hierarchical way, by chunking lower level features together. (e.g. Orbán, Fiser, Aslin, and Lengyel, 2008). Here, we present participants with two sets of patterns composed of simple objects. In one of these sets, the scenes are drawn from a hierarchically structured probability distribution, while in the other one the dependencies are not strictly hierarchic. The subjects' task is to categorize the patterns into one of the two possible categories. If hierarchies are an important aspect of the structure of the environmental systems, to which subjects are attuned, it should be more difficult for the participants to correctly categorize the non-hierarchic objects.

2.2 The Experiment

Participants were presented with images formed by twelve geometric shapes (figure 2.1) and were instructed to categorize them as either food or poison. Whether a certain image pattern is truly food or poison was initially unknown to participants, so that they needed to learn the diagnostic features by trial and error. Every time they responded “food” they were given feedback (“correct” or “incorrect”). There was no feedback when they responded “poison”. Image patterns were sampled from probabilistic graphical models (a graphical representation of the joint distribution of the features in the image), specifically directed acyclic graphs (i.e. Bayes nets; (Pearl, 2001; Bishop, 2006)), designed to meet certain criteria. The Bayes nets had 12 visible nodes, comprising the image stimuli, and 10 hidden nodes (figures 2.2 and 2.3).

These hidden nodes represented the collection of contingencies upon which the nature of the image pattern (food or poison) relied. For example, one hidden node may denote the climate in which a certain fruit is grown, and another hidden node may denote the toxicity of the soil. In our experiment, the individual hidden nodes do not specifically stand for any such condition, rather the entire network of hidden nodes represents a typical network of causations, the end result of which makes the image a food or a poison. There were two sets of images: one sampled from a hierarchic Bayes net and the other from a non-hierarchic Bayes net. The non-hierarchic Bayes network was constructed in such a way that the image patterns sampled from its twelve visible nodes looked similar to the image patterns sampled from the hierarchic network. Note that in this setting, hierarchy is not an all or none property, and the non-hierarchic network still resembles, to some extent, a hierarchic structure (see concluding remarks for discussion).

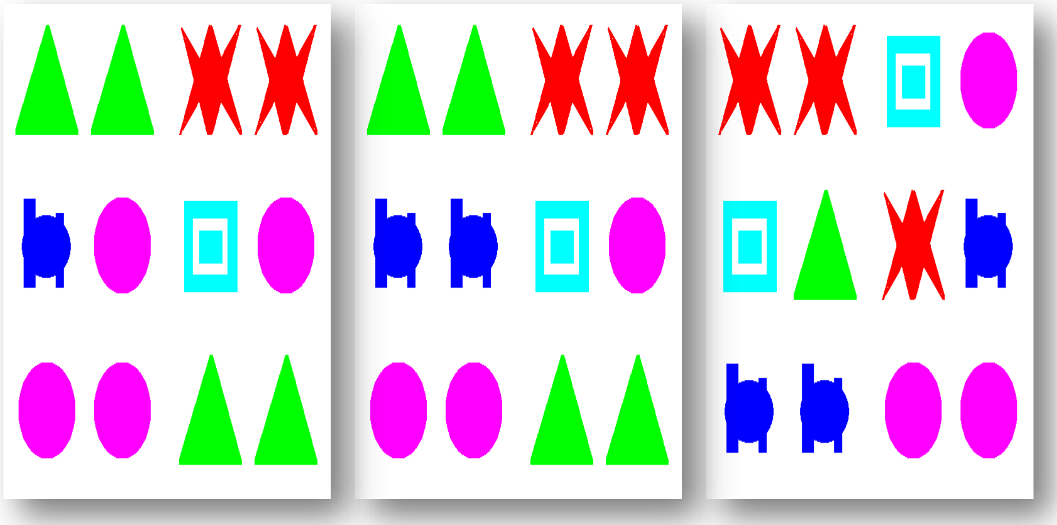


Figure 2.1: participants were instructed to categorize image stimuli into one of the two possible categories, “food” or “poison”. In one condition. The stimuli were generated according to a hierarchic structure. On the top two example images from this condition are presented which were designated as food items. In the other condition, images were generated according to a non-hierarchic structure. On the bottom two example food images from this condition are presented.

2.3 The Non-Hierarchic Case

For stimuli that are generated by a set of non-hierarchic causes, several factors may impair participants’ performance. First, in the environment that participants are familiar with (the real world), causal structures are usually hierarchic. For instance, toxicity of the fruit is a feature formed by several lower level features (lower level merely in the hierarchic sense), such as the molecular structure of the soil, acidity of precipitation, ripeness (fruits that are too ripe are more prone to corruption), etc. We expected, therefore, that participants would try to utilize their existing hierarchic representation of the environment in learning the patterns, and that the mismatch between those representations and the causal structure behind the patterns would impair their performance. At the same time, non-hierarchic representations are more expensive to compute, and should add to the impairment of learning.

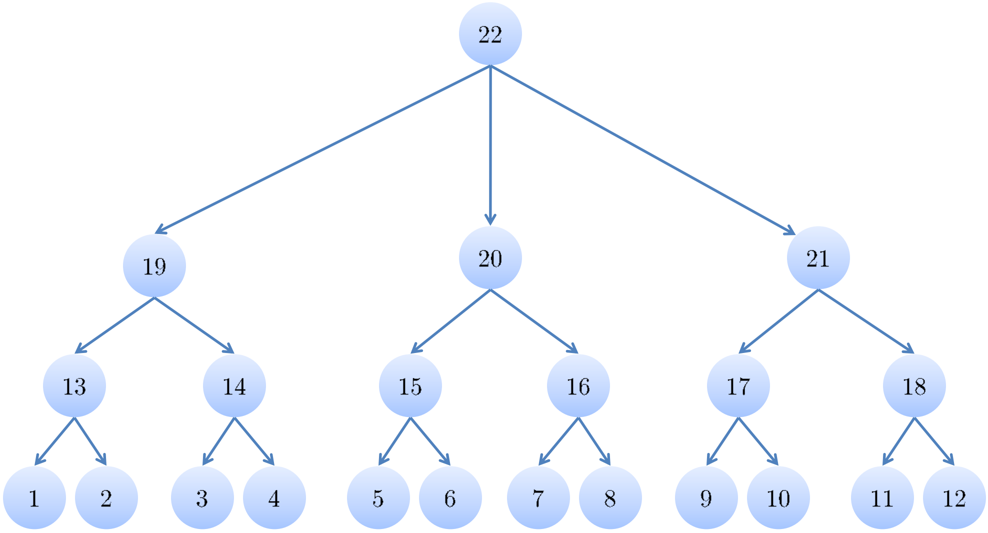
Furthermore, following the premise of statistical learning, participants are trying to learn the probability of a certain image pattern being associated with either food or poison: $Pr\{F = food|I\}$, where F denotes the nutrition content (i.e., food or poison) and I is the image pattern. The pattern consisted of twelve elements (the geometric shapes). Let us call them e_i. Therefore, $I = (e_1, ..., e_{12})$.

Keep in mind that even though the category of I is determined by nodes that are not directly observable, the effect of those hidden nodes must be accessible through the visible nodes, I itself. In fact, if the hidden nodes had no visible manifestation, learning the diagnostic features would be impossible. Therefore, observing that a subset of I, say, $(e_k, ..., e_n)$ has a particular value (e.g. 'star', 'star', 'triangle',...) counts as evidence in inferring the value of a certain hidden node. Ultimately it is the values of these hidden nodes that make the fruit food or poison. In the hierarchic condition, e_i are related to each other in groups that interact locally (figure 2.5, top). For example, (e_1, e_2) are grouped together under the same hidden node; a hidden node from the second level of hierarchy, e_{13}. Similarly, two neighboring hidden nodes from the second level, e_{13} and e_{14}, are grouped together under a hidden node from the third level, e_{19}, and so on. In this situation, the values of the hidden nodes can be inferred in a straightforward manner by observing neighborhood clusters of the visible nodes. For instance, suppose participants have learned that the nutrition content of image patterns can be inferred based on the value of the first hidden node in the third level of hierarchy, e_{19}. The value of this particular node is reflected in the visible nodes e_1 through e_4. Therefore, learning the required diagnostic feature amounts to learning the values of these four nodes. (Note that participants need not have explicit knowledge of the hierarchy. All they need to do is learn implicitly that certain configurations of e_1 through e_4 have a high correlation with poison or food).

In contrast, there is no such straightforward relationship in the non-hierarchic condition. First of all, visible nodes do not interact locally. For example, even though e_5 through e_8 are located close to each other, their features are contingent on hidden nodes which do not directly interact (figure 2.5, bottom). Furthermore, the statistical dependence may have a complicated structure: whereas in the hierarchic condition e_5 through e_8 ultimately depend on one hidden node, e_{19}, in the non-hierarchic condition e_6 and e_7 are governed by both e_{20} and e_{22}, while e_8 depends on e_5, and e_5 is conditionally independent of other nodes (i.e., its values do not depend on the values of the other nodes). The point is that even though there still exists a network of hidden causes that could in principle be used to infer the category of the stimuli, the more complicated structure of dependencies makes such inference more difficult to perform.

2.4 Procedures

Eight participants (four male and four female) took part in the experiment. Each participant performed both the hierarchic and the non-hierarchic conditions in a randomized order. Each condition consisted of 200 trials. It took each participant between fifteen to thirty minutes to complete the experiment. Images were



(a)

Figure 2.2: Graphical representation of the statistical dependencies in the hierarchic condition

(a)

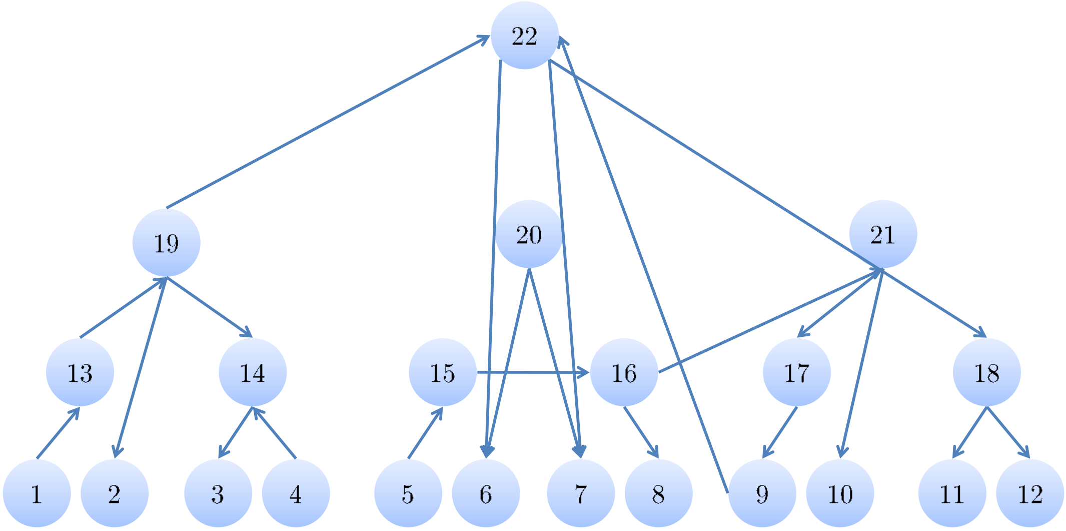


Figure 2.3: Graphical representation of the statistical dependencies in the non-hierarchic condition

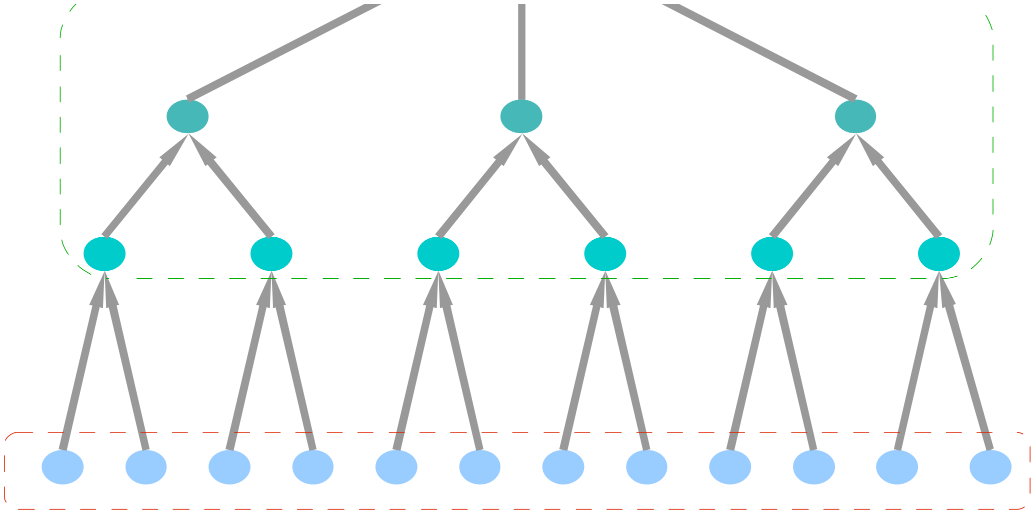


Figure 2.4: Images are sampled from the visible nodes (red dashed line) of the Directed Acyclic Graphs. Hidden nodes (green dashed line) represent the network of causes that determine whether the image is food or poison.

presented on a computer screen using the Psychophysics tool box (Brainard, 1997) running under Matlab. In each trial, an image pattern was presented on the screen and participants had to respond by pressing either “Y”, meaning they believed the stimulus was a food item, or “N” otherwise. There was no time constraint. The next stimulus appeared on the screen immediately after the participants’ response. For each condition of the experiment, the participants initially started with 100 points–their remaining “life”. For every “poison” item accepted, they lost 5 points; for every “food” item they gained 5 points. The last five “food” items that were correctly categorized were displayed at the bottom of the screen. Thus, feedback on the participants’ choice was provided in the form of correct or incorrect only when they responded “Y”.

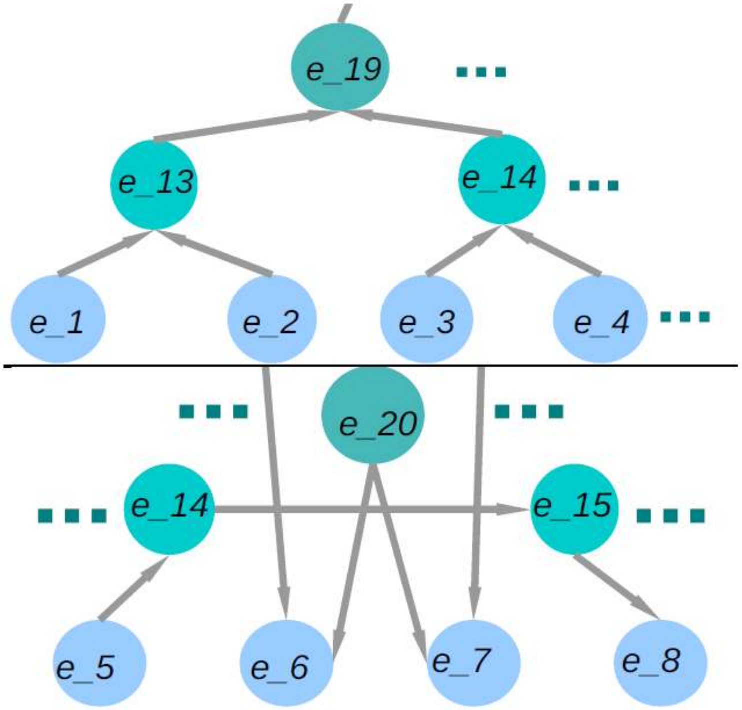


Figure 2.5: In the hierarchic condition proximal nodes' values are related locally (top) and their causal structure is more straightforward. In contrast, in the non-hierarchic condition (bottom) proximal nodes do not necessarily interact locally, and their causal structure is more complex

2.5 Results

Performance was measured as the percentage of correct classifications for each participant in each condition. On average, participants performed 79% correct in the hierarchic condition compared to 63 % correct in the non-hierarchy condition (figure 2.6). This difference in performance is statistically significant as confirmed by the nonparametric Kruskal–Wallis rank sum test ($\chi^2 = 104.91, df = 1, p < 2.2\times10^{-16}$). We also fit a linear mixed model to the data using the `lmer` procedure

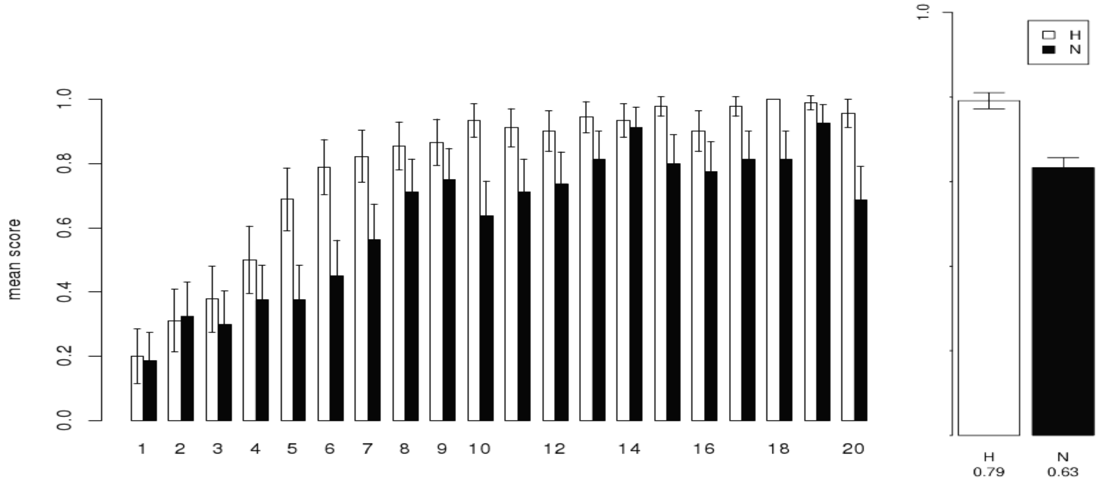


Figure 2.6: *Left*: Comparison of performance in the hierarchic (white bars) versus non-hierarchic (black bars) over 200 trials. Error bars represent 95% confidence limits. Y axis shows mean accuracy of categorization over blocks of 10 trials. *Right*: participants' performance measured as the mean percentage of correct classifications in each condition. H: Hierarchy, N: Non-Hierarchy.

(Bates, 2005). A binomial logit-link linear mixed model fit to the scores yielded a significant effect of condition ($z = 9.85, p < 2.2 \times 10^{-16}$). To explore the effect of gradual learning, we added trial number (in increments of 10) as an independent variable to the linear mixed model. In this analysis, the main effect of condition became n.s., the effect of trial number and the interaction between trial number and condition were both highly significant ($z = 17.96, p < 2 \times 10^{-16}$, and $z = 7.267, p < 3.67 \times 10^{-13}$, respectively; see figure 2.6).

2.6 Concluding Remarks

There are several ways in which a structure can differ from a hierarchy. For example, links can skip levels, or the direction of the causation can be reversed. Consequently, further experiments are required to pin down the effect of each of them. Furthermore, the distinction between a hierarchy and a non-hierarchy is not all or none; rather it is a graded property, with perfect hierarchy at one extreme and heterarchy at the other extreme. We have been unable, however, to find a standard measure of the degree of hierarchicality in the existing literature. Developing and motivating such a measure is a topic for future work.

Another issue for future research is the possibility that subjects performed worse in the non-hierarchic condition of our experiment because the patterns in that condition were more complex. We plan to use the information entropy (Shannon, 1949) of the two graphs, as well as other measures of pattern generator complexity, in investigating this possibility. In the present study, we controlled for pattern complexity at the level of the leaves of the graph, by using stimuli that have the same appearance in both conditions.

CHAPTER 3

LEARNING PROBABILISTIC VISUAL HIERARCHIES

Abstract We live in a world where many complex phenomena, from biological systems to large scale institutions, seem to employ hierarchic schemes in the organization of their parts. In particular, in the feedforward cortical visual stream in the mammalian brain, receptive fields at each successive stage are formed by gathering input from multiple units at preceding stages, resulting in an organization that resembles a hierarchy both in anatomy and in function. This organization may be in part influenced by hierarchical structures in the visual world, and it may in turn make it easier for the system to deal with visual stimuli that are hierarchically organized, as compared to those that are less so. To test the latter prediction, we conducted three behavioral experiments that investigated the influence of the latent structure of visual scenes on their classification. Participants were presented with stimuli drawn from underlying generative probability distributions whose degree of hierarchicality was controlled and had to learn to classify the stimuli as either “food” or “poison.”. We found that participants learn and perform better when the structure of the underlying distribution is more hierarchical[1].

3.1 Introduction

Many natural and artificial phenomena exhibit hierarchical structure, in which elements at one level of organization form larger units at a higher level. The fundamental structure of condensed matter, for instance, is hierarchical: stuff is

[1] This chapter is based on an article co-authored with Shimon Edelman.

organized in molecules, which consist of atoms, which in turn consist of hadrons and leptons, etc. Likewise, many human institutions, such as the military, are organized hierarchically. Visual scenes and objects, too, are often described in hierarchical terms: a face has two eyes, a mouth, and a nose, with the eye having a finer structure of its own, and so on.

The apparent hierarchical structure of visual categories may be in the eye of the beholder, but it may also be due to a combination of external factors, namely, the organizational advantage that hierarchical structures offer and the computational advantage enjoyed by a cognitive system that attunes itself to a hierarchically organized environment. With regard to the former, it has been observed that hierarchically organized natural systems are likely to be less susceptible to perturbations and also to evolve faster than competitors (Simon, 1973, 2002). At the same time, a cognitive agent in the business of survival, which must keep track of the statistics of its environment, can do so more efficiently if it organizes the variables in question hierarchically, a strategy used often in computer vision (Selfridge, 1958; Epshtein, Lifshitz, and Ullman, 2008) and natural language processing (Solan, Horn, Ruppin, and Edelman, 2005; Teh, Jordan, Beal, and Blei, 2006a).

Indeed, the cortical visual pathway in the mammalian central nervous system is organized hierarchically (Felleman and Van Essen, 1991; Modha and Singh, 2010; Kaiser and Hilgetag, 2010). Anatomically, the feedforward visual stream is characterized by a convergence of neuronal projections in a many-to-one fashion. For instance, a simple cell in the primary visual cortical area V1 receives a weighted sum of the outputs of several lateral geniculate nucleus (LGN) cells, each of which in turn is fed by the outputs of several retinal ganglion cells

(Hubel and Wiesel, 1968). At the next stage, the outputs of several simple cells converge on each complex cell. Functionally, too, the visual stream is organized hierarchically, with the receptive fields of cells at each stage computed by combining the characteristics of the receptive fields of cells at the preceding stage (Marr and Hildreth, 1980).

How does the anatomical/functional hierarchy of the cortical visual processing pathway affect the way we perceive the world? Given that this structure may have evolved in the first place in response to certain objective characteristics of the environment, one may expect that the visual system would be better at dealing with stimuli whose categorical structure is itself hierarchically organized (Fiser and Aslin, 2005; Orban, Fiser, Aslin, and Lengyel, 2008), even if this structure is latent, that is, hidden in the probabilistic relationships among individual objects. In this paper, we report a series of behavioral experiments that investigated this hypothesis.

Following Bateson and Hinde (1976), we define a hierarchy as a nested collection of items where no item is superior to itself. Formally, hierarchies can be represented as directed graphs in which vertices denote units and edges indicate containment. We presented participants with images drawn from two possible categories, each defined by a joint probability distribution on their features, while controlling the degree of hierarchicality of the joint distribution. Intuitively, a tree structure is more strictly hierarchical if, for instance, fewer of its edges skip levels. In general, our subjects found it easier to learn to categorize images drawn from more hierarchical distributions.

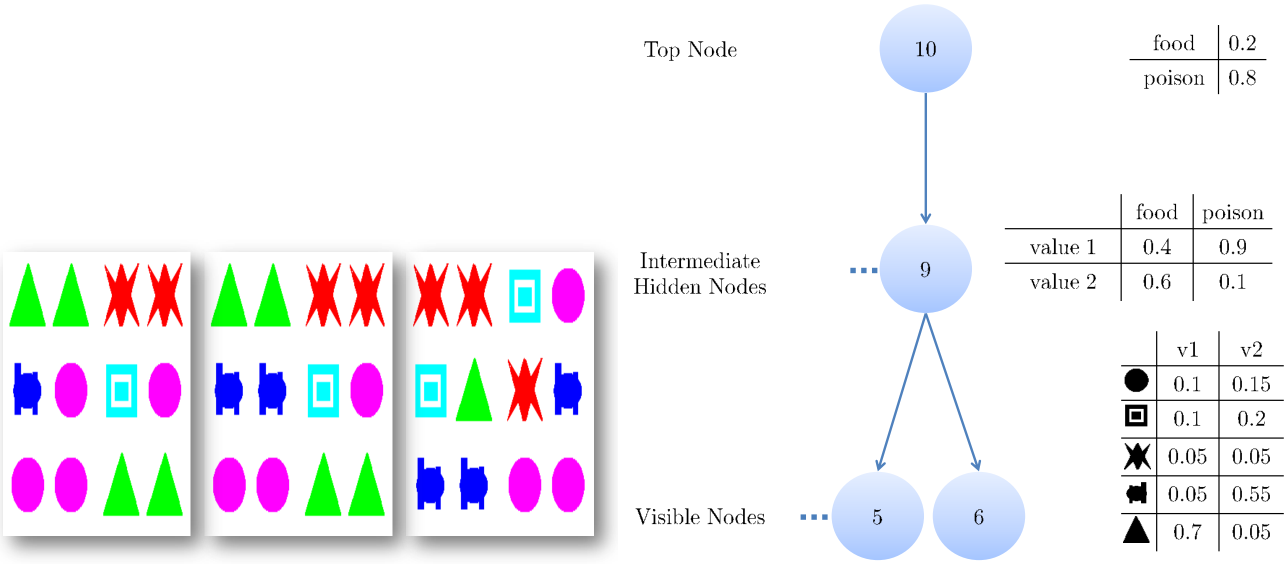


Figure 3.1: *Left:* Three examples of stimuli from Experiment 1. The participants learned to categorize such stimuli as either “food” or “poison” through trial and error. The true nature of the stimuli was determined by a hidden joint probability distribution, which controlled the identity and placement of the features (elementary shapes) comprising the stimuli. *Right:* The structure of the probability distribution from which the stimuli were drawn. The node at the top (hidden) is a random variable determining the category of the stimulus pattern. The bottom nodes correspond to the visible elements that comprise a stimulus. Each random variable at the bottom corresponds to one particular location in the image. At the sampling stage, one of the five possible elementary shapes is selected and used in that location. As an example, the conditional probability tables are shown for the top node (hidden), one intermediate node (hidden), and one visible node. In each table, the left column shows the possible values of the corresponding node; the right column shows their probabilities, given the parent node’s values. Node numbers reflect the actual enumeration used in $H1$, Figure 3.5.

3.2 The experimental approach

In each of the following experiments, we manipulated the degree of hierarchicality of the probability distribution underlying the category structure of the stimuli.

In Experiment 1, there were two conditions, corresponding to hierarchical and non-hierarchical distributions. In Experiments 2 through 5, we controlled the degree of hierarchicality, defined via a measure based on path lengths and degrees of branching. Furthermore, to be able to attribute the difference in performance to the difference in hierarchicality, we equalized the information content of the distributions across conditions.

The stimuli in the experiments were defined by visible and hidden features. The visible features are the elementary shapes that comprise the stimulus image (see Figure 3.1, left, for three examples). The hidden features, corresponding to the structure of the probability distribution (Figure 3.1, right), can be thought of intuitively as environmental factors that are not immediately accessible to the participants, yet contribute to making an item “food” or “poison” (e.g., properties of climate and soil in an area where a certain fruit grows). The category of the stimulus (food or poison) was also hidden and had to be learned by the participants from experience.

The manner in which hidden factors contributed to how the property of being food or poison manifested itself in the image was defined by a probability distribution, which took the form of a conditional dependence graph (Pearl, 2000; see Figure 3.1, right). Once the graph structure and its corresponding distributions are set, a stimulus is generated by drawing a random sample. The value of the top node determines the category of that stimulus (food or poison), and the value of the bottom nodes determines what elements will be used in the image. Each of these bottom nodes correspond to one particular location in the image. Intermediate hidden features are not explicitly coded in the stimuli; rather, they control the structure of the conditional dependence of the visible features on the category. It is the hierarchicality of this structure that was the independent variable of our experiments.

We considered two main factors controlling the hierarchicality of the graph structures. The first factor is how closely it resembles a tree. A tree is a graph in which there is exactly one path connecting any two given vertices (Bondy and Murty, 1976). In our setup, the existence of more than one such path means that the child node depends conditionally on more than one parent node. In other words, it means that the visible features of a stimulus are controlled by several hidden factors. We hypothesized that in this situation it would be more difficult for the participants to learn the diagnostic features. The second factor is the number of children that a parent node has, i.e., the degree of branching. A high degree of branching indicates that the same hidden factor controls several visible features. We hypothesized that this too would make learning correct categorization more difficult.

In each of the experiments, participants were presented with stimuli in the form of arrays of elementary shapes (Figure 3.1, left), and were instructed to categorize them as food or poison. Correct categorization required an understanding (possibly implicit) of the diagnostic features of each category, which had to be learned by trial and error. Diagnostic features were combinations of particular elements (basic shapes) and their locations. For instance, a possible food feature could consist of two stars in the top right corner and a triangle in the bottom left corner, or two squares next to each other anywhere in the image.

Before starting the experiment, participants were handed written instructions, complemented by the experimenter's verbal assistance, that defined their task as learning to categorize image patterns as food or poison. They were told that they would be presented with fruits from an alien planet, and they needed to learn which fruits are good to eat. Further, they were informed that just as on Earth particular combinations of features may make a fruit edible (e.g. an apple that is vibrant red, firm to the touch, and smells good), particular combinations of these alien features (i.e. the geometric shapes making up the stimulus) are the diagnostic features that they should try to learn. They were also instructed that the association between stimuli and their category label was probabilistic and that the same image pattern could have some probability of being food and some probability of being poison. In addition, they were informed that the conditions (blocks of trials, whose beginning and end were displayed on the screen) were independent, and that whatever features they learned for one condition would not transfer to the other.

The Bayes Network Toolbox for Matlab (Murphy et al., 2001) was used in implementing the probabilistic models. The stimuli were presented on a computer screen using Matlab Psychophysics Toolbox (Brainard, 1997). Participants responded to each stimulus by pressing 'Y' for food and 'N' for poison. Each time they responded 'Y' they were given feedback; no feedback was provided when they responded 'N'. To aid the learning process, the last five correctly classified food items were displayed at the bottom of the screen to be compared against each other. There was no time constraint: participants were free to take as much time as they needed. The next stimulus appeared on the screen immediately after the participants' response. To help them keep track of their performance, in the beginning of the experiment they were awarded 100 points; for every poison item accepted, five points were deducted; conversely, five points were gained for every food item accepted. They were given explicit information that this score was only for them to track their performance and would not participate in the data analysis.

3.3 Experiment 1

3.3.1 Stimuli and methods

Experiment 1 consisted of two conditions, hierarchic and non-hierarchic, with 200 trials per condition. Each stimulus was formed by 12 elementary shapes (Figure 3.1, left). The graphs corresponding to the underlying probability distributions in the two conditions are shown in Figure 3.2. The graph on the left shows the hierarchic condition; the one on the right shows the non-hierarchic (actually, less hierarchic; more on this later) condition.

For the hierarchic graph, the distributions of individual nodes were sampled from a Dirichlet distribution with uneven weights, so that particular values of the random variables could be assigned a larger mass (Bishop, 2006). The distributions of the non-hierarchic graph were then trained on samples drawn from the hierarchic one, making the distributions of identically numbered nodes match. In both graphs, the top node's probability mass values were manually set to 20% food and 80% poison.

Eight subjects participated in the experiment. Each participant performed both the hierarchic and the non-hierarchic conditions (blocks) in a randomized order.

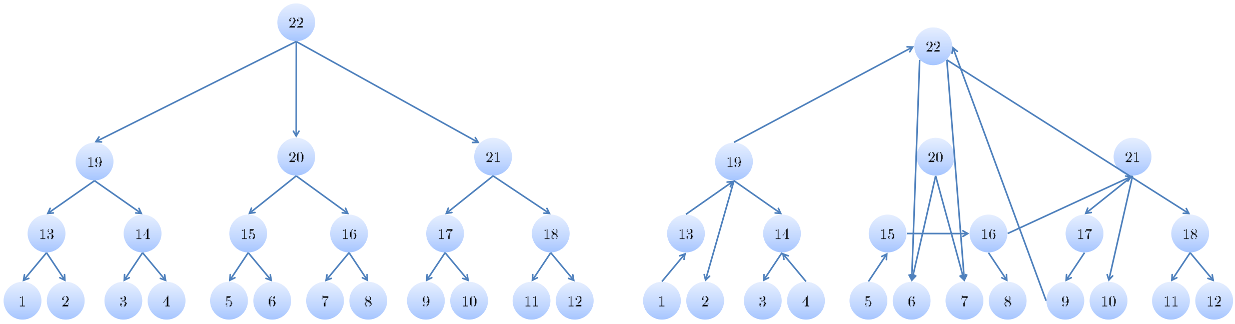


Figure 3.2: Directed acyclic graphs representing the underlying joint probability distributions for the two conditions of Experiment 1. *Left:* hierarchy. *Right:* non-hierarchy. In each graph, the top node denotes the random variable representing the food/poison choice. The bottom 12 nodes denote random variables representing the elementary shapes comprising the stimulus pattern (for some examples, see Figure 3.1, left). Starting with the top node, a sample is drawn from each random variable conditioned on the sample drawn from its parent(s). The values drawn from each of the last 12 random variables are matched to their corresponding shape and assembled into a complete stimulus pattern.

3.3.2 Results

Performance was measured as the rate of correct responses in each condition. The mean correct rate was 79% in the hierarchic condition, compared to 63% in the non-hierarchy condition (Figure 3.3, left). The difference between conditions was statistically significant, as revealed by a nonparametric Kruskal-Wallis rank sum test ($\chi^2 = 104.91,\ df = 1,\ p < 2.2 \times 10^{-16}$). Because significant effects can still be rendered n.s. when all the random effects are considered jointly (Baayen, 2008), we also fit a linear mixed model to the data using the lmer procedure (Bates, 2005). A binomial logit-link linear mixed model fit to the scores yielded a significant effect of condition ($z = 9.85,\ p < 2.2 \times 10^{-16}$).

To explore the effect of gradual learning, we added trial number (in incre-

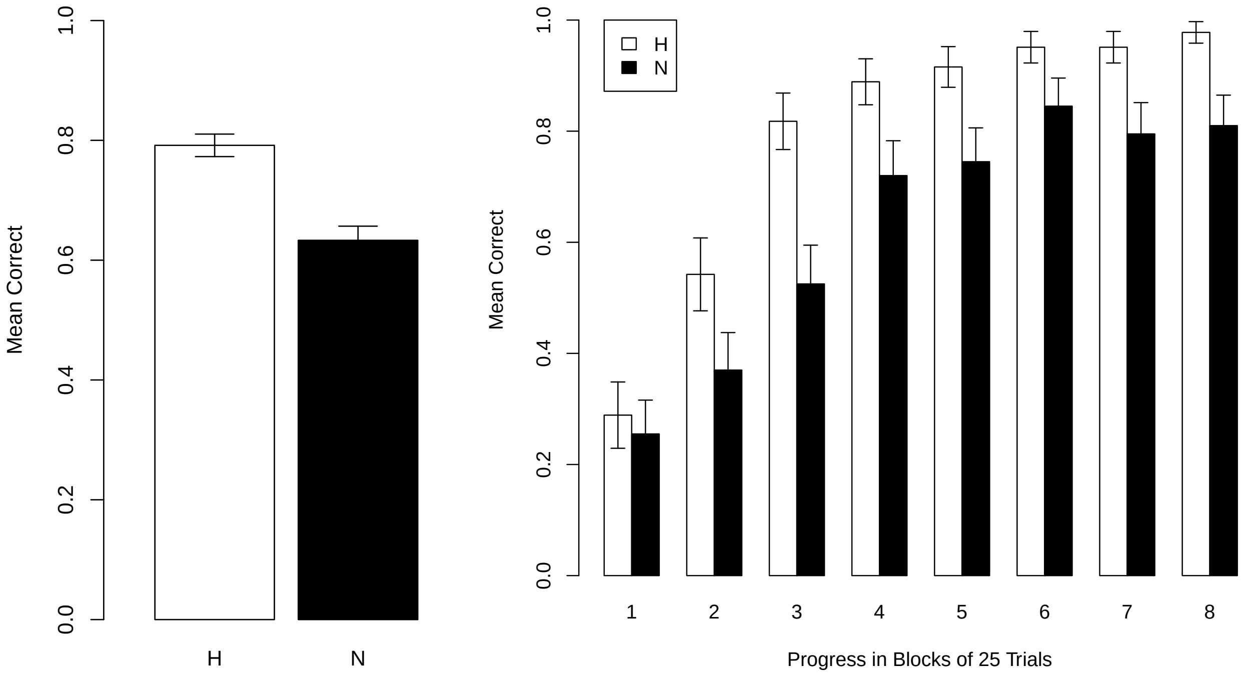


Figure 3.3: Mean categorization performance in Experiment 1. *Left:* mean performance for each of the two conditions. Error bars represent 95% confidence intervals. *Right:* the development of performance throughout the experiment. The 200 trials of each condition are here grouped into eight blocks of 25 trials.

ments of 25) as an independent variable to the linear mixed model. In this analysis, the main effect of condition became n.s.; the effect of trial number and the interaction between trial number and condition were both highly significant ($z = 17.96$, $p < 2\times10^{-16}$, and $z = 7.267$, $p < 3.67\times10^{-13}$, respectively; see Figure 3.3, right).

3.3.3 Discussion

The results of Experiment 1 suggest that probabilistic (generative) hierarchical structure of composite visual objects affects their learning. However, because this experiment involved only two levels of hierarchicality, it could not reveal its graded effects, if any.

To be able to attribute the difference in performance to the difference in the underlying hidden structure of the categories, in this experiment we made sure that the actual stimuli were visually identical across conditions; only their structured probabilistic association with the category labels differed. However, the effect that we found could still be due to the underlying distribution in the non-hierarchic case being harder to learn. This can happen, for instance, when the distributions are close to uniform and do not favor particular values. To address these issues we devised a second experiment.

3.4 Experiment 2

Experiment 2 consisted of four conditions that were equally complex, but varied quantifiably in how hierarchic they were. This design allowed us to determine whether the effect of hierarchicality on categorization performance is graded. Complexity across the conditions was equalized by controlling the joint entropies of the distributions. Because no published standard approach to measuring hierarchicality could be found in the literature, we introduce our own formulation in the following section.

3.4.1 Stimuli and methods

Figures 3.5 and 3.4 show, respectively, the four graphs used to generate the stimuli in the four conditions of Experiment 2 and a sample of the resulting stimuli. We aimed to assign a score, H, to each graph that would reflect how

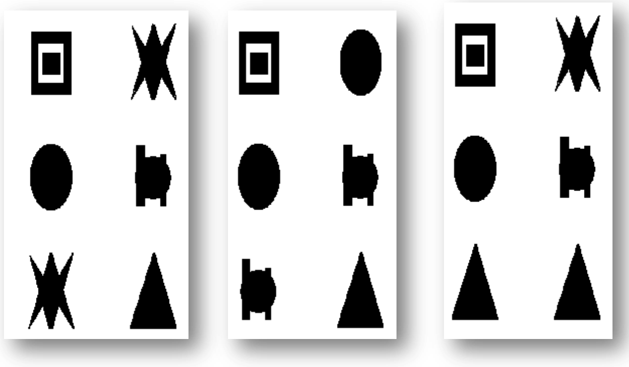


Figure 3.4: A sample of stimuli from Experiment 2.

closely it resembled a strict hierarchy. The main two factors that we wanted this formulation to emphasize were the degree of branching and the resemblance of the graph to a tree. Accordingly, we defined H as follows:

$$H = \frac{\sum_j |S_j - K|}{N} + \delta \tag{3.1}$$

where S_j is the length of the shortest path between the j^{th} vertex and the top node, and K is the nominal length of such path (in an l-level hierarchy, for a vertex on the l^{th} level, $K \doteq l$). For the six visible nodes of the four graphs, K is set to the statistical mode of the path lengths. Note that only $H1$ and $H2$ have mid-level nodes, and the contribution of these nodes to H is zero. Also note that because the graphs are directed, the choice of the top node is unambiguous. N is the total number of vertices in the graph, and δ is a term penalizing the average branching degree of the graph:

$$\delta = \frac{\sum d_j}{N \times K} \tag{3.2}$$

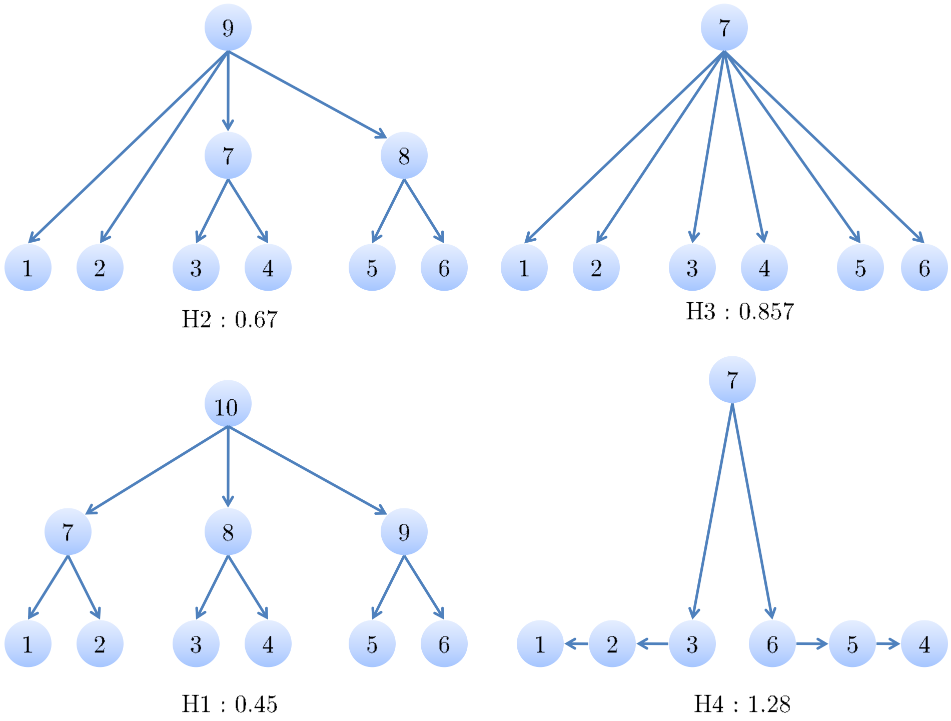


Figure 3.5: The four directed acyclic graphs representing the underlying joint distributions for the four conditions of Experiment 2. The degree of hierarchicality, as defined by eq. 3.1, is listed for each graph.

where d_j is the degree of branching (i.e., the number of edges fanning out) of node j. We introduced δ into H to distinguish between a tree with few children per node from a tree with many children per node. Consider a *linear* tree where each node has exactly one node as opposed to a *flat* tree where the number of children per node approaches infinity. The first half of eq. 3.1 assigns identical scores to both of these trees. By incorporating δ, the score of the flat tree is made higher, as per its deviation from hierarchicality. The purpose of dividing by K is normalization, so that larger trees do not get penalized unfairly.

To illustrate the above points, we follow step by step the computation of H for the $H3$ condition. The mode of path lengths in this case is 1, so $K = 1$. For all six nodes, $S_j = 1$. This means that the first half of eq. 3.1 is equal to $\sum |1 - 1|/7 = 0$.

The average degree of branching is $\frac{6}{7}$ normalized by $K = 1$. Thus, the total score is $H = 0 + \frac{6}{7}$.

Once a measure of hierarchicality is assigned, we equalize the joint entropy– over all the random variables involved including, top, middle and bottom leaves– of the underlying distributions for the four conditions, to avoid a situation in which differences in learnability are due merely to differences in the statistical complexity of the distributions. For that purpose, we use Shannon's entropy for a multivariate distribution (Shannon, 1949):

$$H(X_1, ..., X_n) = -\sum_{x_1} ... \sum_{x_n} P(x_1, ..., x_n) \log[P(x_1, ..., x_n)]$$

We began by fixing the values of top node probability at 0.2 and 0.8 respectively for food and poison, so that 20% of the stimuli generated from each graph would be food and the rest poison. The distributions of the remaining nodes were then sampled from a Dirichlet distribution. Finally, the joint entropies for all graphs were equalized by an optimization procedure (gradient descent on the empirical gradient of the entropy).

The test setup of Experiment 2 was similar to that of Experiment 1, with a few changes. Instead of twelve color shapes, each stimulus consisted of six black and white shapes. To keep the length of the experiment reasonably short, each participant performed only two out of the four possible conditions. These two conditions were chosen at random for each participant. Each condition contained of 100 trials. A total of 56 subjects participated in the experiment (23, 30, 26, and 33, for $H1$, $H2$, $H3$, and $H4$ respectively).

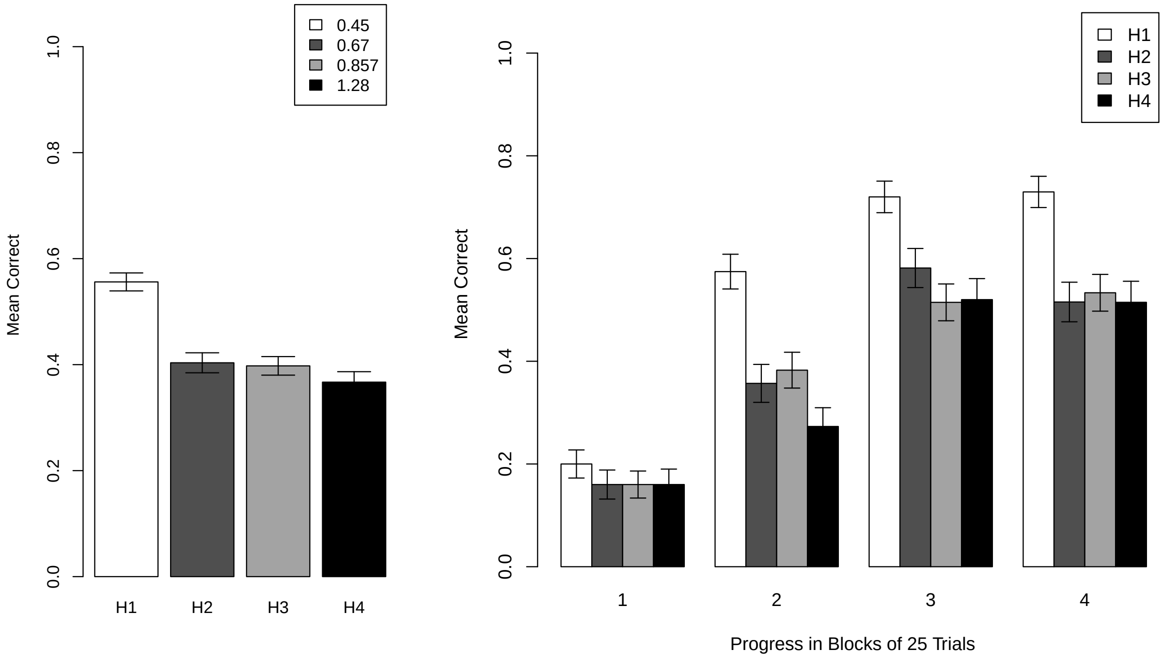


Figure 3.6: Mean categorization performance in Experiment 2. *Left:* mean performance for each of the two conditions. Error bars represent 95% confidence intervals. *Right:* the development of performance throughout the experiment. The 100 trials of each condition are here grouped into four blocks of 25 trials.

3.4.2 Results

The mean performance levels in the four conditions of Experiment 2 were 54.9%, 40.3%, 39.7%, and 36.7% for $H1$ through $H4$ respectively (see Figure 3.6). Regression analysis revealed significant trends for the measure of hierarchicality: for a linear regression, $R^2 = 0.302$, $F = 49.02$, $df = 110$, and $p < 2.11 \times 10^{-10}$; for quadratic regression, $R^2 = 0.413$, $F = 39.98$, $df = 109$, and $p < 9.513 \times 10^{-14}$ (Figure 3.7).

To quantify the progression of learning, we broke down each condition into

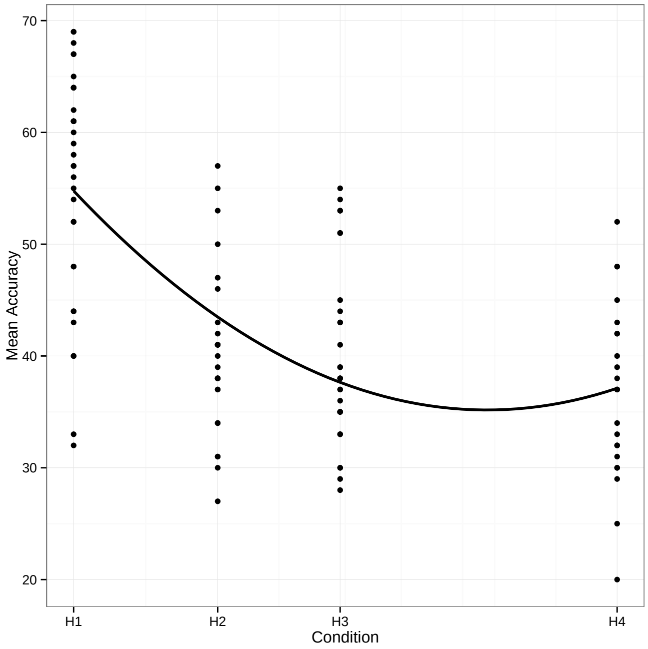


Figure 3.7: Quadratic regression of performance on hierarchicality in Experiment 2 ($R^2 = 0.413$, $F = 39.98$, $df = 109$, $p < 9.513 \times 10^{-14}$).

blocks of 25 trials and performed regression on two variables: Condition and Progression. For the first degree polynomial regression, we obtained $R^2 = 0.437$, $F = 434.9$, $df = 1117$, and $p < 2.2 \times 10^{-16}$. For the second degree polynomial, we obtained $R^2 = 0.494$ $F = 219.4$, $df = 1114$, and $p < 2.2 \times 10^{-16}$. Furthermore, Condition and Progression, as well as their quadratic forms, were all significant with $p < 8.01 \times 10^{-12}$, but their interaction was n.s. ($p = 0.0848$).

As in Experiment 1, we also fit a binomial logit-link mixed effects model to the data, using the lmer procedure. This analysis confirmed that both Condition and Progression effects were significant with $p < 1.1 \times 10^{-9}$.

3.4.3 Sensitivity to the parameters involved in defining H

The definition of hierarchicality that we introduced earlier in this paper (eq. 3.1) involves implicit free parameters that control the relative contributions of different measures of the graph structure to the value of H. It is reasonable to ask how these degrees of freedom affect the statistical picture of the subjects' performance painted by our experiments. We addressed this question in two ways.

First, we treated H itself as a parameter and used the mixed effects analysis to obtain values of the Akaike Information Criterion (AIC) and the Bayesian Information criterion (BIC) for each value of H. These measures of model fit were then used in an empirical gradient descent search, to determine which value in the four-dimensional space of hierarchicality measures (corresponding to $H1$ through $H4$) resulted in the best fit. The search algorithm was run 10 times, for 1000 iterations each. We repeated the mixed effect regression analysis for the 10 best and the 10 worst values of H found in this manner. In all 20 cases, the effect of H on the subjects' performance was still significant, at least with $p < 10^{-3}$.

In a second analysis, we parameterized H explicitly as a simplex:

$$H = \alpha \frac{\sum_j |S_j - K|}{N} + (1 - \alpha) \frac{\sum d_j}{N \times K} \tag{3.3}$$

and searched for the optimal α by descending on the empirical gradient of AIC. The value of α minimizing AIC was found to be $\alpha > 15$. However, the minimum value of AIC was smaller only by 7 units than the value that resulted from our original definition of H, signifying a negligible improvement in the goodness of fit.

In conclusion, these two analyses suggest that the significance of the results of our experiment does not depend on the specifics of our formulation of H.

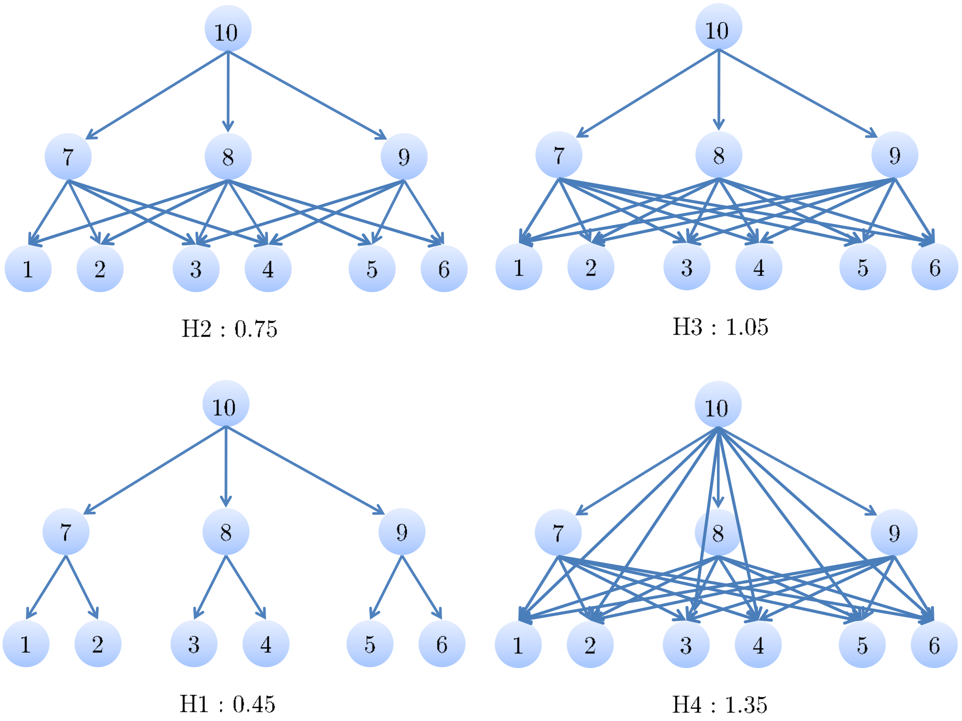
3.4.4 Discussion

Although all the regression analyses in this experiment yielded a significant dependence of categorization performance on hierarchicality, an inspection of the plots indicates that this effect may have been driven mostly by the subjects' better performance in the $H1$ condition. Indeed, when we omitted $H1$ from the analyses, the effect of H became non-significant. To address this issue, we conducted a third experiment, in which we looked more closely at the intermediate values of H.

3.5 Experiment 3

3.5.1 Stimuli and methods

In Experiment 3, we generated a new set of stimuli with more evenly distributed values of hierarchicality: the successive levels differed from one another by 0.3 units (Figure 3.8). The statistical complexity of the new stimuli was equalized to that of the stimuli in Experiment 2. Other than the new stimulus set, everything in the setup of Experiments 2 and 3 was the same. Fifteen subjects participated in this experiment.



H2 : 0.75

H3 : 1.05

H1 : 0.45

H4 : 1.35

Figure 3.8: The four directed acyclic graphs used to generate the stimuli for the four conditions of Experiment 3, in which the levels of hierarchicality were set to be equally far apart (0.3 units). The degree of hierarchicality (eq. 3.1) for each graph is indicated.

3.5.2 Results

The mean performance in the four conditions was 46.8%, 43.2%, 38.3%, and 21.2% for $H1$ through $H4$ respectively (Figure 3.9). The linear regression of performance on hierarchicality (Figure 3.10) was significant: $R^2 = 0.576$, $F = 40.36$, $df = 28$, and $p < 7.12 \times 10^{-7}$. The quadratic term was n.s. To make sure that the significance of the results was not driven by just one condition, we carried out an analysis with $H4$, the condition in which the performance differed the most from the others, omitted. The outcome was still significant: $R^2 = 0.210$, $F = 7.408$, $df = 23$, and $p < 0.01$.

To quantify the progression of learning, we broke down each condition into

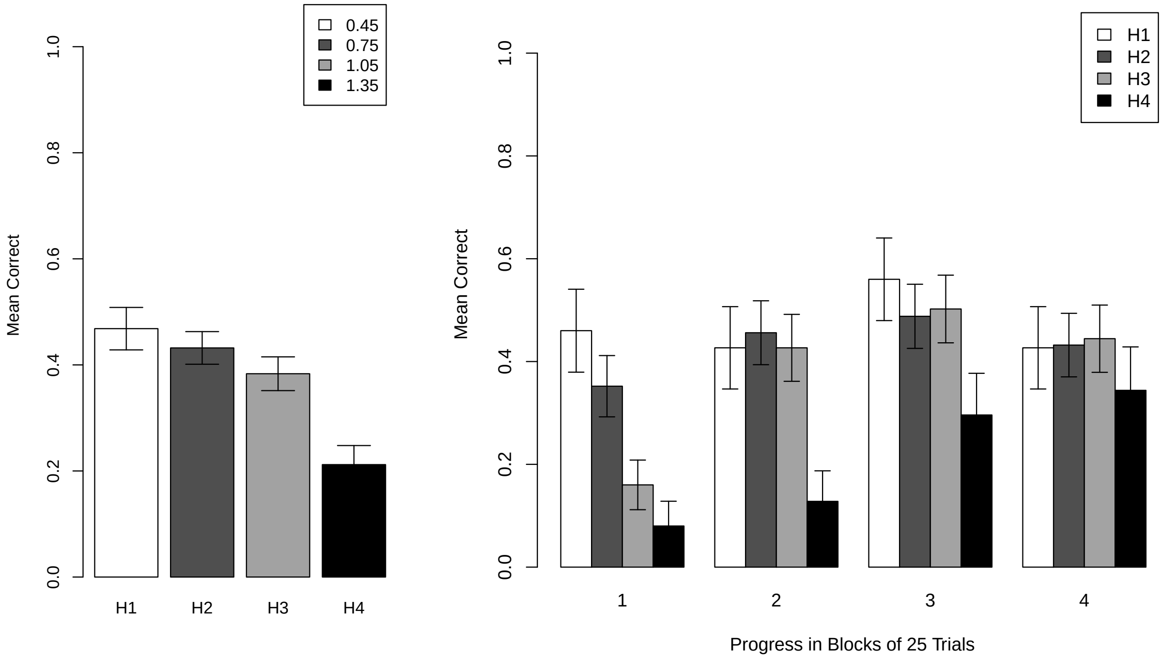


Figure 3.9: Mean categorization performance in Experiment 3. *Left:* mean performance for each of the two conditions. Error bars represent 95% confidence intervals. *Right:* the development of performance throughout the experiment. The 100 trials of each condition are here grouped into four blocks of 25 trials.

blocks of 25 trials. A regression over the resulting two fixed effects, Progression and Condition, resulted in a linear fit with $R^2 = 0.173$, $F = 206.2$, $df = 1947$, and $p < 2.2 \times 10^{-16}$, and a quadratic fit with $R^2 = 0.201$ $F = 99.1$, $df = 1944$, and $p < 2.2 \times 10^{-16}$. For the linear fit, both Condition and Progression effects were significant, $p < 3.86 \times 10^{-15}$. For the quadratic fit, Progression but not Condition was significant. The interaction was significant, $p < 0.0073$.

As before, we also performed a mixed effect analysis using the lmer procedure on the entire data (see section 3.4.2 for details), which confirmed that both Condition and Progression were significant, $p < 1.87 \times 10^{-10}$.

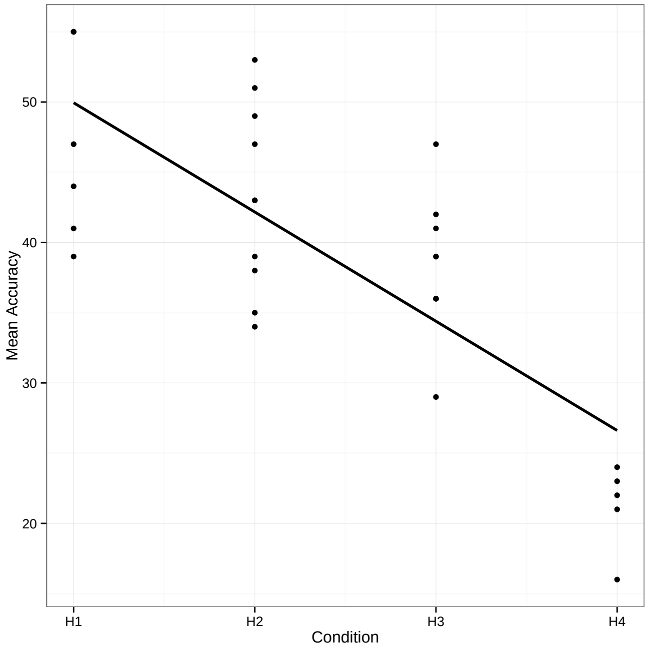


Figure 3.10: Linear regression of performance on hierarchicality in Experiment 3; $R^2 = 0.576$, $F = 40.36$, $df = 28$, $p < 7.12 \times 10^{-7}$.

3.5.3 Experiments 2 and 3 together

Taken together, the levels of hierarchicality in the different conditions of Experiments 2 and 3 span a broad range of values. Pooling together the data from these experiments and performing a linear regression yielded $R^2 = 0.361$, $F = 80.66$, $df = 140$, and $p < 1.62 \times 10^{-15}$ (figure 3.11). A quadratic regression resulted in $R^2 = 0.396$, $F = 47.27$, $df = 139$, and $p < 2.2 \times 10^{-16}$.

3.5.4 Discussion

The collective results of the three experiments reported so far suggest a significant interaction of hierarchicality with learnability. However, there still are a number of factors that these experiments fail to control for. First and foremost, our

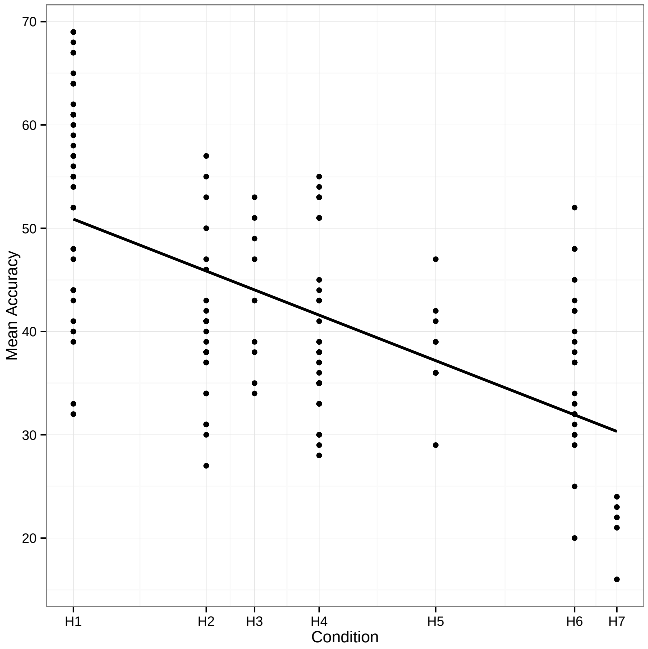


Figure 3.11: Combined data from Experiments 2 and 3. Taken together, the eight conditions of these experiments correspond to seven distinct measures of hierarchicality ($H1$ was equal to 0.45 in both experiments). This plot shows a linear regression for the pooled data: $R^2 = 0.361$, $F = 80.66$, $df = 140$, $p < 1.62 \times 10^{-15}$.

measure of hierarchicality incorporates two different properties of the graphs, namely, degree of branching and skipping levels. This raises the question of what the contribution of each property to learnability is. Although our analyses in section 3.4.3 do address this issue to some extent, we decided it would be ideal to treat each term directly and in isolation.

Furthermore, in our experimental protocol, we presented subjects with the last five correctly categorized food items to aid feature detection. However, in natural settings seldom do agents have access to an explicit record of past encounters with stimuli.

Finally, in all the previous experiments the distribution of stimuli were 20%

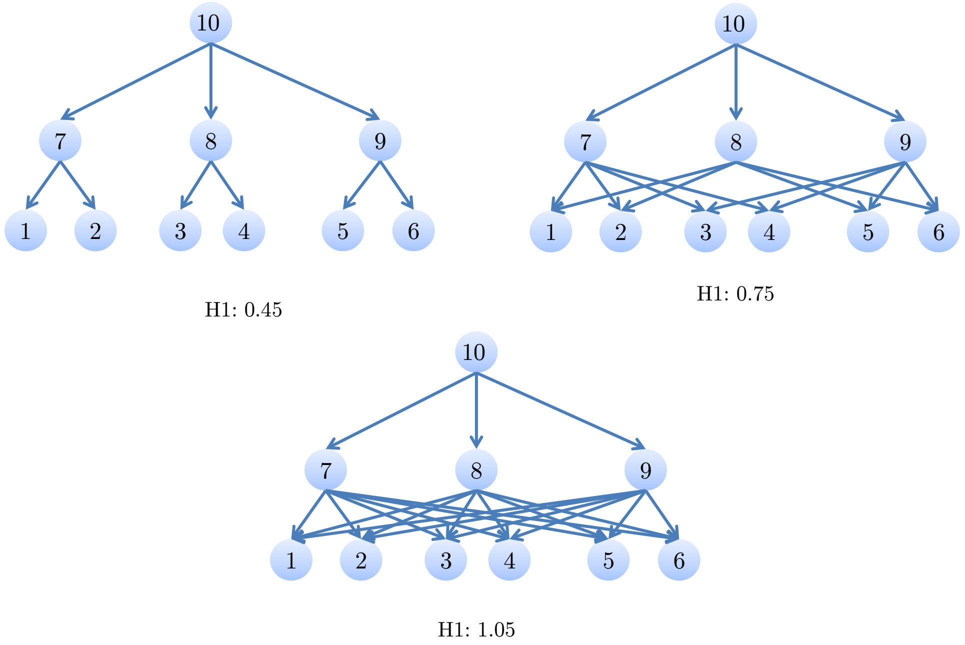
food and 80% poison; we needed further control runs to find out whether this particular distribution was confounding the results. Consequently, we devised experiments 4 and 5 to remedy the above mentioned shortcomings.

3.6 Experiments 4 & 5

For experiments 4 and 5 we devised two sets of generative distributions with varying measures of hierarchicality. In one set, called *skipping*, the contribution of the penalty term for degree of branching was fixed, and all the difference in structure was due to links skipping levels (figure 3.13). In the other set, called *branching*, the penalty term for skipping levels was fixed while the degree of branching changed (figure 3.12). This setting allowed us to study the effect of each of the terms (equations 3.1 and 3.2) in our measure of hierarchicality in isolation. Each set contained three conditions with varying measures of hierarchicality.

Furthermore, unlike the previous experiments, we relied on Kullback-Leibler divergence as the measure of complexity, which we minimized using random search over the parameter space.

As before, in experiment 4 the distribution of food and poison items were 20% and 80%, respectively, and the last five correctly categorized food items were displayed for the subjects. In contrast, in experiment 5, food and poison were evenly distributed (50% each), and the only display was the target stimulus. Each condition consisted of 150 trials. The rest of the experimental paradigm was the same as the previous three.



H1: 0.45

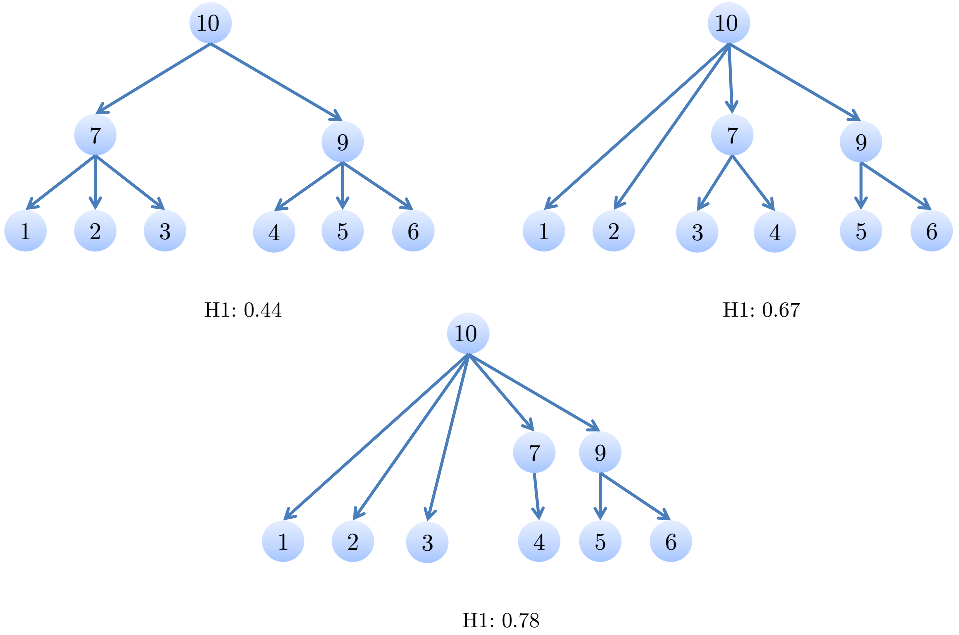
H1: 0.75

H1: 1.05

Figure 3.12: Graphical representation of probability distributions underlying the *branching* set of conditions in experiments 4 and 5. σ, the first term in the measure of hierarchicality that penalizes skipping levels is the same for all three graphs. But each one has a different value of δ, the second term which penalizes degree of branching. H1: $\sigma = 0; \delta = \frac{9}{20}$; total = 0.45 – H2: $\sigma = 0; \delta = \frac{15}{20}$; total = 0.75 – H3: $\sigma = 0; \delta = \frac{21}{20}$; total =1.05

3.6.1 Results of experiment 4

16 subjects participated in the *branching* set of conditions, and 29 in the *skipping* set. Mean performance of the *branching* set was 54%, 51%, and 42% respectively for H1, H2, and H3 (figure 3.14). Linear regression of performance on hierarchicality for *branching* (Figure 3.15) shows the effect is significant with $R^2 = 0.186$, $F = 8.99$, $df = 31$, and $p = 0.005043$. The quadratic effect was n.s. Mean performance of the *skipping* set was 53%, 51%, and 41% respectively for H1, H2, and H3 (figure 3.16). Linear regression of performance on hierarchicality for *skipping*



H1: 0.44

H1: 0.67

H1: 0.78

Figure 3.13: Graphical representation of probability distributions underlying the *skipping* set of conditions in experiments 4 and 5. δ, the second term in the measure of hierarchicality that penalizes degree of branching is the same for all three graphs. But each one has a different value of σ, the first term which penalizes skipping levels. H1: $\sigma = 0; \delta = \frac{8}{18}$; total = 0.44 – H2: $\sigma = \frac{2}{9}; \delta = \frac{3}{9}$; total = 0.67 – H3: $\sigma = 0; \delta = \frac{8}{18}$; total =0.78

(Figure 3.17) shows the effect is significant with $R^2 = 0.1466$, $F = 11.14$, $df = 57$, and $p = 0.00148$. The quadratic effect was n.s.

3.6.2 Results of experiment 5

14 subjects participated in the *branching* set of conditions, and 20 in the *skipping* set. Mean performance of the *branching* set was 50%, 51%, and 50% respectively for H1, H2, and H3 with a n.s. effect: $p = 0.73$. Mean performance of the *skipping* set was 49%, 51%, and 49% respectively for H1, H2, and H3 with a n.s. effect:

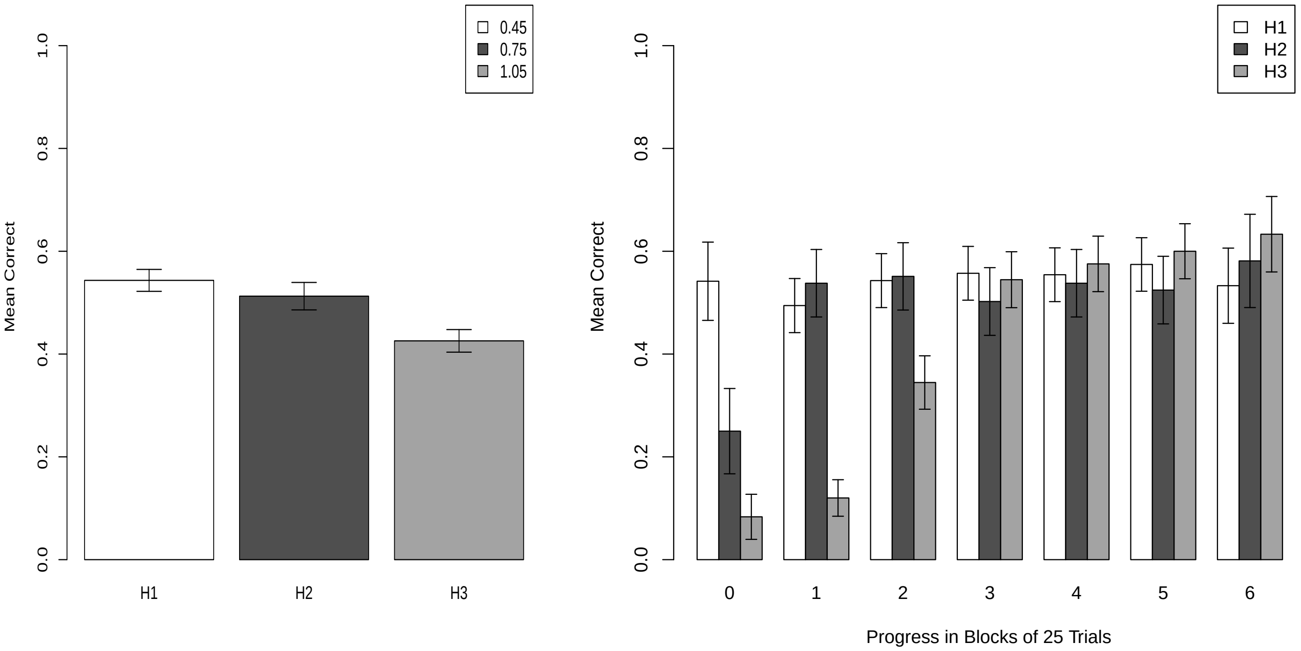


Figure 3.14: Mean categorization performance in *branching* conditions of experiment 4. *Left:* mean performance for each of the three conditions. Error bars represent 95% confidence intervals. *Right:* the development of performance throughout the experiment. The 150 trials of each condition are grouped into six blocks of 25 trials.

$p = 0.88$.

3.6.3 Discussion

While the results of experiment 4 were in line with our previous findings, experiment 5 failed to replicate our earlier observations in either *branching* or *skipping* condition sets, with performances that did not deviate from chance level (50%).

Experiments 4 and 5 shared the same graph structure in their underlying

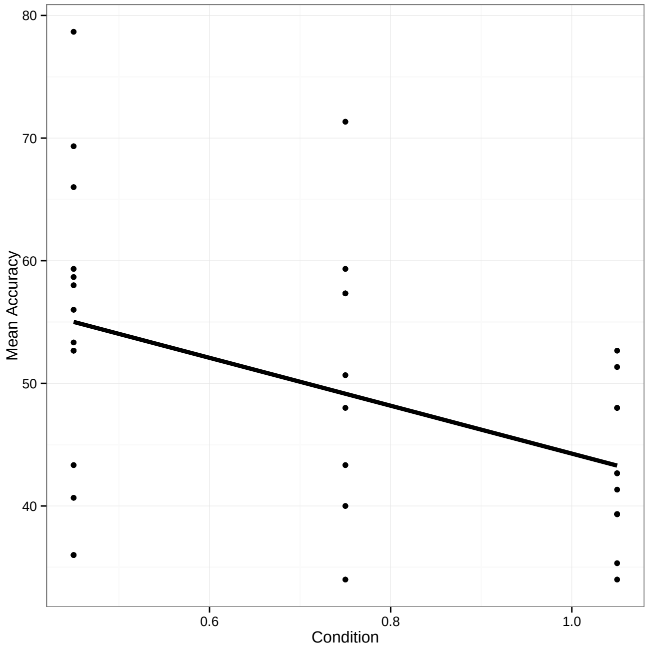


Figure 3.15: Linear regression of performance over condition for the *branching* set of experiment 4: $R^2 = 0.186$, $F = 8.99$, $df = 34$, and $p = 0.005043$

distributions. However, the distribution of food and poison stimuli were different (20% food – 80% poison in experiment 4 vs. 50% food – 50% poison in experiment 5). In order to attribute the different performances in the two experiments to their different distributions of target stimuli, we would have to consider that the specific parameters of the experimental setup favored certain distributions over others. This is unlikely because the particular aspects of our setup have undergone several changes since experiment 1, but the effect has remained significant through experiment 4.

The more likely culprit is the absence of the five exemplars that subjects were shown to aid detection of diagnostic features. We are planning a new set of experiments to help narrow down the factors involved in the impaired performance on experiment 5.

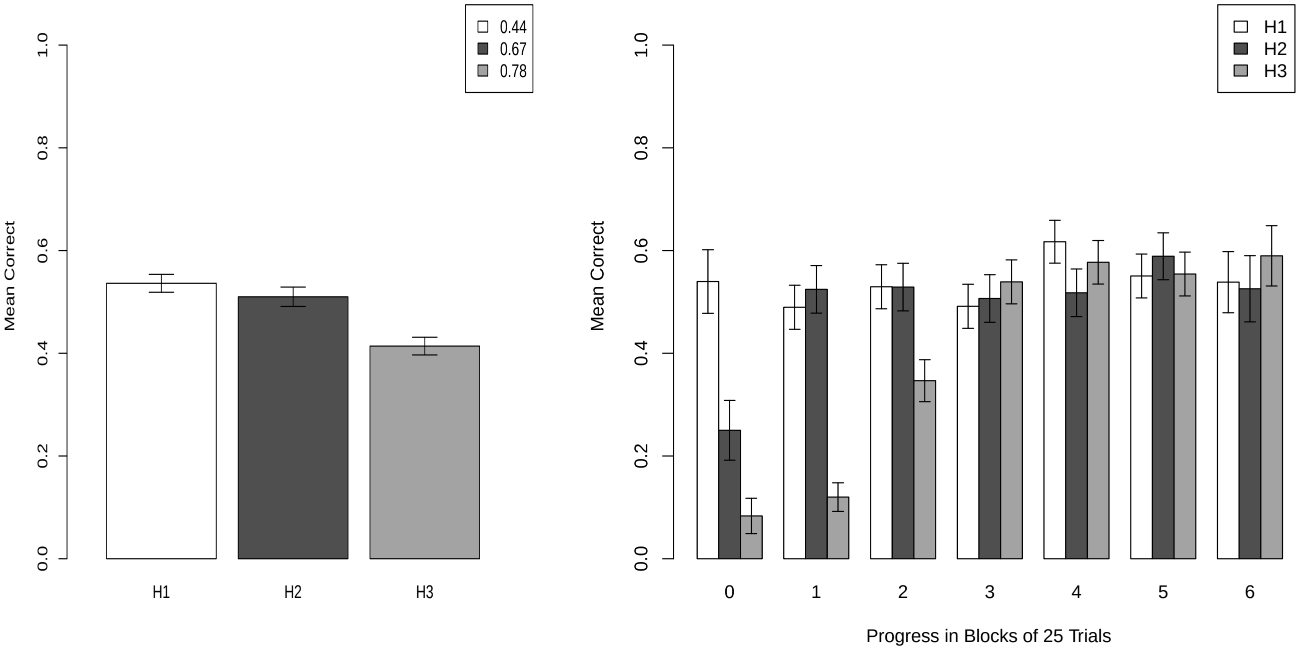


Figure 3.16: Mean categorization performance in *skipping* conditions of experiment 4. *Left:* mean performance for each of the three conditions. Error bars represent 95% confidence intervals. *Right:* the development of performance throughout the experiment. The 150 trials of each condition are grouped into six blocks of 25 trials.

On the other hand, the results of experiments 4 did uphold our previous findings in both *branching* and *skipping* conditions, suggesting that *i*) both terms involved in our formulation of hierarchicality interact with subject performance in learning, and *ii*) equalizing complexity using the more strict metric of Kullback-Leibler divergence yields similar results to using joint entropy.

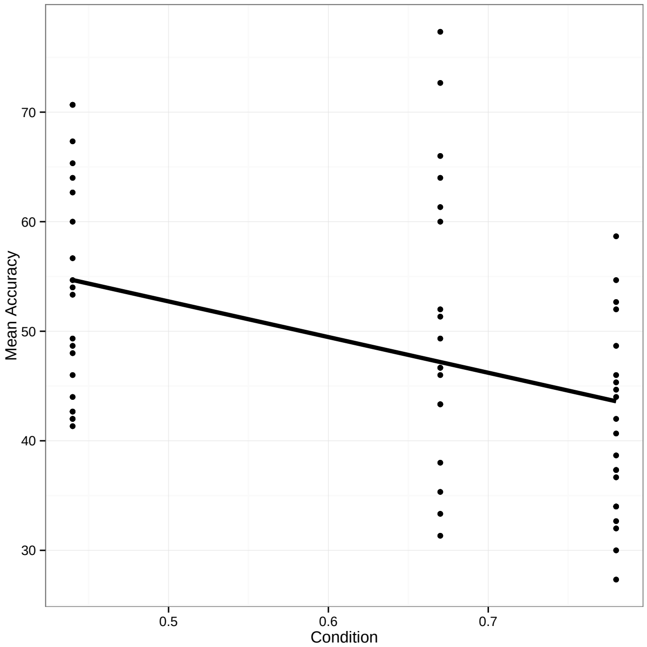


Figure 3.17: Linear regression of performance over condition for the *skipping* set of experiment 4: $R^2 = 0.1466$, $F = 11.14$, $df = 57$, and $p = 0.00148$

3.7 General discussion

In the studies described in this paper, we set out to investigate the effects of hidden probabilistic hierarchical structure of visual stimuli on their categorization. In four experiments, we found that stimuli generated from distributions that are quantitatively more hierarchical are indeed easier for subjects to categorize.

In Experiment 1, there were only two conditions, which differed considerably in the degree of hierarchicality of the stimuli. This difference was reflected in the participants' less accurate performance in the less hierarchic condition. Experiment 2, in which there were four levels of hierarchicality, suggested that the dependence of performance on hierarchicality may be graded. This finding was supported by the results of Experiment 3, in which hierarchicality was more carefully controlled. Experiment 4 suggested that each of the two penalty terms in H contributes to the observed effect, and that the statistical significance of results are maintained after replacing joint entropy with Kullback-Leibler divergence. The results of experiment 5 were non-significant.

One may speculate regarding the possible explanations for the dependence of the ease of learning and categorization on the hierarchical structure underlying the stimuli. In the environment that the participants are familiar with (the real world), causal structures are often hierarchical. For instance, the toxicity of a fruit may depend in complex ways on features that are hidden from the observer, such as the genetic makeup of the plant, the composition of the soil, the acidity of precipitation, and the degree of ripeness (fruits that are unripe or are too ripe may be inedible). It can be expected, therefore, that participants bring their implicit expectation of hierarchicality to the artificial learning environment of the experiment, and that their performance suffers when this expectation is not fully met.

What mechanisms could mediate this effect? A visible feature may only be diagnostic to the extent that it reveals the values of hidden features. In a hierarchical structure, a spatially compact cluster of visible features may depend on the same hidden cause, which would then be easier to infer than in the case of a less hierarchical hidden structure, in which the correlated feature could be spatially far-flung.

Finally, from the vantage point of statistical learning, less hierarchical representations may be more expensive to compute, due to the curse of dimensionality (Hastie, Tibshirani, Friedman, and Franklin, 2005). Tracking the statistics of the environment amounts to recording co-occurrences of various events. On the one hand, the more events are kept track of, the more powerful the inferences that can be performed. On the other hand, the resources required for such bookkeeping grow exponentially in the number of events to be recorded. One solution is to group together events that co-occur often enough, and treat them as one — in other words, to form a hierarchy of events. In an environment where events do not lend themselves to such representation, maintaining detailed statistics can be too expensive, which would make learning more difficult.

Acknowledgments

We thank David Field for discussions in the early stages of this project and Amy Chen, Danielle Czirmer, and Max Levinson for their help with running the experiments. The results of Experiment 1 were first reported in the Proceedings of the 33rd Cognitive Science Society Conference (2011).

CHAPTER 4

SIMILARITY, KERNELS, AND THE FUNDAMENTAL CONSTRAINTS ON COGNITION

Abstract The kernel trick, which was devised in statistical learning theory as a shortcut to expensive high dimensional computations, has broad and constructive implications for the brain sciences. Regarding the kernel not so much as an implicit map onto a high dimensional space, rather, as a measure of similarity that offers low dimensional and low complexity decision rules, opens up several venues for their application in cognitive information processing. Here we specify four fundamental constraints that must be met by any nervous system that learns from the statistics of its world, and discuss how kernel-like neural computations can serve perceptual learning and decision making, while observing those constraints[1].

4.1 Motivation and plan

The concept of similarity is widely used in psychology. Historically, in a philosophical tradition dating at least back to Aristotle, it has served as a highly intuitive, unifying slogan for a variety of phenomena related to categorization. Here's how Hume put it in the *Enquiry* (1748):

> ALL our reasonings concerning matter of fact are founded on a species of Analogy, which leads us to expect from any cause the same events, which we have observed to result from similar causes. Where the

[1] This chapter is based on an article co-authored with Rajeev Raizada and Shimon Edelman.

> causes are entirely similar, the analogy is perfect, and the inference, drawn from it, is regarded as certain and conclusive. [...] Where the objects have not so exact a similarity, the analogy is less perfect, and the inference is less conclusive; though still it has some force, in proportion to the degree of similarity and resemblance.

In the past century, psychologists have turned similarity into a powerful theoretical tool, most importantly by honing the ways in which similarity can be grounded in multidimensional topological or metric representation spaces (see Osgood, 1949 for an early example) or in situations where a set-theoretic approach may seem preferable (Tversky, 1977).

Sometimes criticized as too loose to be really explanatory (e.g., Goodman, 1972), the concept of similarity has eventually been given a mathematical formulation, including a derivation from first principles of the fundamental relationship between similarity and generalization (Shepard, 1987). These mathematical developments have solidified similarity's status as a theoretical-explanatory construct in cognitive science (Ashby and Perrin, 1988; Medin, Goldstone, and Gentner, 1993; Goldstone, 1994; Edelman, 1998; Tenenbaum and Griffiths, 2001; for a recent review, see Edelman and Shahbazi, 2012).

In the present paper, we explore the parallels between the psychological construct of similarity and its recent mathematical treatment in the neighboring discipline of machine learning, where a family of classification and regression methods has emerged that is based on the concept of a kernel (Schölkopf and Smola, 2002). Insofar as kernels (described formally in a later section) involve the estimation of distances between points or functions (Jäkel, Schölkopf, and Wichmann, 2008, 2009), they are related to similarity. At the same time, there seems to be a deep rift between the two.

On the one hand, similarity-based learning and generalization has long been thought to require low-dimensional representations, so as to avoid the so-called “curse of dimensionality” (Bellman, 1961; Edelman and Intrator, 1997, 2002), as well as to promote the economy of information storage and transmission (Joliffe, 1986; Roweis and Saul, 2000). Moreover, as no two measurements of the state of the environment are likely to be identical, some abstraction is necessary before learning becomes possible, which calls for information-preserving dimensionality reduction (Edelman, 1998, 1999). On the other hand, the best-known kernel methods, based on the Support Vector Machine idea (Cortes and Vapnik, 1995; Vapnik, 1999), involve a massive increase in the dimensionality of the representation prior to solving the task at hand.

We attempt to span this rift by seeking a common denominator for some key ideas — and, importantly, their mathematical treatment — behind similarity and kernels. In service of this goal, we first identify, in section 4.2, four fundamental constraints on cognition, having to do with (i) measurement, (ii) learnability, (iii) categorization, and (iv) generalization. In section 4.3, we then show that while on an abstract-functional or task level these constraints appeal to the concept of similarity, on an algorithmic computational level they call for the use of kernels. Section 4.4 revisits some standard notions from the similarity literature in the light of this observation. In section 4.5.1, we illustrate the proposed synthesis by pairing the methods that it encompasses with a range of cognitive tasks. In section 4.5, we suggest some ways in which these methods can be used to further our understanding of computation in the brain. Finally, section 4.6 offers a summary and some concluding remarks.

4.2 Fundamental constraints on cognition

4.2.1 A fundamental constraint on measurement

Perception in any biological or artificial system begins with some measurements performed over the raw signal (Edelman, 2008, ch.5). In mammalian vision, for instance, the very first measurement stage corresponds to the retinal photoreceptors transducing the image formed by the eye's optics into an array of neural activities. The resulting signal is extensively processed by the retinal circuitry before being sent on to the rest of the brain through the optic nerve.

Effectively, a processing unit at any stage in the sensory pathway and beyond "sees" the world through some measurement function $\phi(\cdot)$. Importantly, the measurement process is, at least in the initial stages of development, *uncalibrated*, in the sense that the precise form of the measurement function is not known — that is, not explicitly available — throughout the system. For example, the actual, detailed weight, timing profiles, and noise properties of the receptive field of a sensory neuron are implicitly "known" to the neuron itself (insofar as these parameters determine its response to various types of stimuli), but not to any other units in the system. Indeed, for the usual developmental reasons, those parameters vary from one neuron to the next in ways that are underspecified by the genetic code shared by all neurons in an organism.

Even if the system learns to cope with this predicament (as suggested by some recent findings; Pagan, Urban, Wohl, and Rust, 2013), such learning can only be fully effective if driven by calibrated stimuli, which are by definition not available in natural settings. Moreover, a system that relies on learning, be it as part of its development or as part of its subsequent functioning, it must either (i) simultaneously learn the structure of the data and its own parameters, or (ii) learn the former while being insensitive to the latter.

These considerations imply the following fundamental challenge:

M_0 Any system that involves perceptual MEASUREMENT is confronted with unknowns that it must learn to tolerate or factor out of the computations that support the various tasks at hand, such as learning and categorization (see Tables 4.5 and 4.4).

To the best of our knowledge, this is the first statement of the measurement constraint in the literature. On a somewhat related note, Resnikoff (1989) observed that the general measurement uncertainty principle, as formulated by Gabor (1946), is important for understanding perception. For a recent review of uncertainty in perceptual measurement and the role of receptive field learning under this uncertainty, see (Jurica, Gepshtein, Tyukin, and van Leeuwen, 2013).

4.2.2 Three fundamental constraints on learning

In learning tasks, the need to generalize from labeled to unlabeled data (in supervised scenarios) or from familiar to novel data (in unsupervised scenarios) imposes certain general constraints on the computational solutions (Geman, Bienenstock, and Doursat, 1992). Although here we focus on categorization, where the goal is to learn class labels for data points, these constraints apply also to regression, where the goal is to learn a functional relationship between independent and dependent variables (Bishop, 2006).

According to the standard formulation in computational learning science, the problem of learning reduces, on the most abstract level of analysis, to probability density estimation (Chater et al., 2006). Indeed, the knowledge of the joint probability distribution over the variables of interest allows the learner to compute, for a query point, the value of the dependent variable, given the observed values (measurements) of the independent variables.[2] This basic insight serves as a background for the present discussion.

In this section, we briefly discuss the constraints that apply to (i) the computation of *similarity* among stimuli, (ii) to the *dimensionality* of representation spaces, and (iii) to the *complexity* of the decision surfaces.

Similarity

Estimating the *similarity* among stimuli is arguably the most important use to which sensory data could be put. As mentioned in the introduction, similarity constitutes the only principled basis for generalization, and therefore for any non-trivial learning from experience (Hume, 1748; Shepard, 1987; Edelman, 1998; Edelman and Shahbazi, 2012). Following Shepard (1987), we therefore observe:

S_0 The fundamental challenge facing any system that is expected to generalize from familiar to unfamiliar stimuli is how to estimate SIMILARITY over stimuli in a principled manner.

[2] In this sense, the joint probability distribution over the representation space is the most that can be known about a problem. To know more — for instance, to know the directions of causal links between variables — observation alone does not suffice: one needs intervention (Steyvers, Tenenbaum, Wagenmakers, and Blum, 2003; Pearl, 2009), a topic which is beyond the scope of the present survey.

Dimensionality

Given a representation space for which similarity has been defined, a straightforward and surprisingly effective approach to generalize category labels is to assign to the query point a label derived from its nearest neighbor(s) (Cover and Hart, 1967). Importantly, this approach is nonparametric, in that no particular functional form is assumed for the underlying probability distribution function.

To ensure uniformly good generalization, the nearest neighbor approach requires that the representation space be "tiled" with exemplars, so that any new query point would fall not too far from familiar ones. This requirement gives rise to the so-called "curse of dimensionality" (a concept first formulated in the context of control theory; Bellman, 1961): the tiling of the problem representation space with examples, and with it learning to generalize well, becomes exponentially less feasible as the dimensionality of the space grows. Hence, the following constraint:

D_0 The fundamental challenge facing any learning system is how to reduce the effective DIMENSIONALITY of the problem so as to allow learning from the typically sparse available data (Intrator and Cooper, 1992; Edelman and Intrator, 1997).

We remark that the effective dimensionality of a problem need not be the same as its nominal dimensionality, which is inherited from the measurement or representation space in which the problem arises. In particular, the parametric form of the decision or regression surface (or, more generally, of the underlying joint probability distribution) may be known independently, in which case the effective dimensionality is determined by that form. Likewise, in the support vector approach to classification (Cortes and Vapnik, 1995), the nominal dimensionality, which is equal to the number of features (dimensions of the representation space), is raised drastically when the problem is remapped into a new space that affords linear discrimination, yet its effective dimensionality is determined by the typically very small number of "support vectors" — key data points that determine the width of the classifier margin. More on this below and in section 4.3.4.

Complexity

If the parametric form of the probability distribution is known, or if a particular form is adopted as a working hypothesis, subject to evaluation, then the focus in assuring good generalization shifts from the nominal dimensionality of the representation space to the number of parameters that need to be learned. As noted by Cortes and Vapnik (1995), it was R. A. Fisher (1936) who first formalized the two-class categorization problem and derived a Bayesian-optimal solution to it in the form of a quadratic discriminant function, which he recommended to approximate by a linear discriminant in cases where the number of data points is too small relative to the dimensionality of the measurement space — a very common predicament, known in learning theory as the problem of sparse data. Since then, the idea of keeping the number of parameters small — including opting whenever possible for the smallest number of parameters for a given problem, as afforded by the linear classifier — proved to be a manifestation of a very general principle that governs generalization from data.

Support for Fisher's recommendation comes from converging ideas in the theory of information and computation (Solomonoff, 1964), the Minimum Description Length Principle or MDL (Rissanen, 1987), nonparametric estimation

(Geman et al., 1992), regularization theory (Evgeniou, Pontil, and Poggio, 2000), and statistical learnability theory based on the concept of Vapnik-Chervonenkis (VC) dimension (Blumer, Ehrenfeucht, Haussler, and Warmuth, 1989), which is in turn founded on empirical risk minimization (Vapnik, 1999). This latter approach, which leads to Support Vector Machines, is described by Vapnik as follows: "To generalize well, we control (decrease) the VC dimension by constructing an optimal separating hyperplane (that maximizes the margin). To increase the margin we use very high dimensional spaces."

On the face of it, the second desideratum identified by Vapnik — a high-dimensional representation space — runs counter to principle D_0 identified earlier. However, as we shall see in section 4.3.2, it is made unproblematic by the so-called "kernel trick," which ensures that the *effective* dimensionality of a problem approached in this manner is dictated by the number of data points, rather than by the number of intermediate representation-space "features," which need never be computed explicitly (Jäkel, Schölkopf, and Wichmann, 2007). The windfall from this mathematical fact allows us to focus on the first part of Vapnik's statement:

C_0 The fundamental challenge facing any categorization system is how to remap the problem it faces into a space where it becomes a matter of low COMPLEXITY — preferably, linear — discrimination.

4.3 Kernel-based methods

The four fundamental constraints listed above — M_0 (measurement), S_0 (similarity), D_0 (dimensionality), and C_0 (complexity) — are simultaneously satisfied by a family of computational approaches based on the concept of *kernel*.

4.3.1 Origins

In different mathematical contexts the term kernel can be used in somewhat different senses. The general common theme is that a kernel defines an equivalence relation between a subset of elements in the domain of application. For instance, in abstract algebra the kernel of a homomorphism ϕ from a group G to another group with identity element e is the subset of G that gets mapped to e, namely $\mathrm{Ker}\,\phi := \{x \in G \,|\, \phi(x) = e\}$.[3] In particular, in linear algebra the kernel of a linear map over a vector space is the subspace that gets mapped to zero (Gallian, 2010).

The word "kernel" can also refer to the characteristic property of certain mathematical operations. Used in this sense, kernel means seed, nucleus, or the central aspect of the operation with respect to which it is defined. For instance, in the general theory of stochastic processes a Markov kernel characterizes the state transitions of the process, similar to a Markov transition matrix.[4]

The particular sense of kernel that we are interested in here is the one that arises in the theory of Reproducing Kernel Hilbert Spaces, or RKHS, where a

[3] A *homomorphism* is a structure-preserving map between two algebraic structures, such as groups or fields. A *group* is a set of elements together with an operation that combines any two elements to form a third, which also belongs to the set, while satisfying the four group axioms: closure, associativity, identity, and invertibility.

[4] Our search for an underlying relationship between the two senses of the term yielded no results, suggesting that this terminology may be an artifact of translation from the German *kern*.

kernel is defined as a symmetric function of two arguments:

$$k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$$

$$k(x, y) = k(y, x)$$

Furthermore, k must be positive semi-definite. For the purposes of this paper this means that for a given sample set $x_1, x_2, ...x_n$, the matrix K whose ij^{th} entry denotes $k(x_i, x_j)$ must be positive semi-definite.[5]

In keeping with the common theme of the term kernel, this definition induces an equivalence relation whereby pairs of points are assigned by the kernel function the same value of distance: $\{(x, y) \,|\, k(x, y) = d\}$. In the rest of this paper, we use the term kernel in this particular sense.

The interesting property of kernels that makes them useful in pattern recognition and machine learning is that they are equivalent to the inner product of some, possibly unknown, function of their arguments.[6] Formally, $k(x, y) = \langle \phi(x), \phi(y) \rangle$, for some $\phi(\cdot)$. We explain this concept in more detail in the following sections.

4.3.2 The "kernel trick"

In this section we first demonstrate application of the kernel trick in an example, and then discuss it in more detail for the general case.

[5] Formally, the matrix K is positive semi-definite if for any vector v the product $v^T K v$ is greater than or equal to zero.

[6] An *inner product* over a vector space is a map from pairs of vectors to a field (e.g., to the real numbers) that satisfies symmetry (for the complex field, conjugate symmetry), linearity, and positive definiteness.

Simple example of the kernel trick

In the following exposition, we draw on materials from (Schölkopf and Smola, 2002; Balcan, Blum, and Vempala, 2006; Jäkel et al., 2007, 2008, 2009). As an example, consider the problem of classifying objects, represented by points in some multidimensional measurement space, into two or more categories. Information that would support such classification may not be available in individual features (dimensions) of the objects or even in their linear combinations. In such cases, one may resort to the use of a polynomial classifier, whose input features include, in addition to the original dimensions, some or all of their products (Boser, Guyon, and Vapnik, 1992).

For example, suppose the original signal is $x \in \mathbb{R}^2$, and the feature of interest is contained in the product $x_1 x_2$. We can provide the classifier with this feature by mapping the data points from their original space to a new one via $\phi : \mathbb{R}^2 \to \mathbb{R}^3$, $(x_1, x_2) \to (x_1^2, x_2^2, \sqrt{2} x_1 x_2)$. However, in practice one does not usually know the appropriate mapping, ϕ, for a given set of data. Therefore, ϕ is chosen to be very flexible (e.g., a very high order polynomial), so as to accommodate a wide range of possibilities (Jäkel et al., 2007).

Unfortunately the computational cost of such mappings can be prohibitive, particularly when the original data reside in a high-dimensional space (as does any set of megapixel-resolution images); cf. the fundamental constraint D_0. For instance, computing a d-degree polynomial for an N-dimensional x, requires computing $(N + d - 1)!/d!(N - 1)!$ monomial terms. Ideally, one would like to keep the advantages of high-dimensional feature spaces while reducing the cost of working with them. This is where kernel-based approaches come in handy. As stated earlier, a kernel is a nonnegative definite, symmetric function of two arguments, $k(x, y) : \mathbb{R}_x \times \mathbb{R}_y \to \mathbb{R}$. It can be shown that such a kernel corresponds to the inner product of its arguments $k(x, y) = \langle \phi(x), \phi(y) \rangle$ (cf. section 4.3.2 for details on this) modified under some function $\phi(\cdot)$ defined over the original domain, which may be desirable but expensive to compute. Identifying a kernel $k(\cdot, \cdot)$ for a function $\phi(\cdot)$ makes it possible, therefore, to evaluate for a given x and y the inner product $\langle \phi(x), \phi(y) \rangle$ directly, without having first to compute the expensive $\phi(x)$ and $\phi(y)$. For instance, in the above example opting for $k(x, y) = \langle x, y \rangle^2$, we get:

$$x = (x_1, x_2),\ y = (y_1, y_2)$$

$$\phi(x) = (x_1^2, x_2^2,\ \sqrt{2}x_1x_2),\ \phi(y) = (y_1^2, y_2^2,\ \sqrt{2}y_1y_2)$$

$$\langle \phi(x), \phi(y) \rangle = (x_1^2y_1^2 + x_2^2y_2^2 + 2x_1y_1x_2y_2) = \langle x, y \rangle^2 = k(x, y)$$

As long as the use to which the data are put depends only on inner products (as in PCA and SVM), using a kernel allows one to enjoy the advantages of a high-dimensional representation space without having to pay the price of explicit computations in that space. This is known as the "kernel trick" (Bishop, 2006).

The kernel trick in general

The primary motivation for using kernels, at least in the classical machine learning view, is that often the nominal representation of data to be used in learning is not linearly separable, and does not lend itself to the multitude of tried-and-true classifying algorithms that require linear separability for their operation, such as the perceptron, Fisher's linear discriminant, and principal component analysis.

In such cases, one may use a dimension-raising map, $\phi(\cdot)$, to transform data points from their original representation into a higher dimensional one attained by $\phi(\cdot)$, in hopes that they will become linearly separable. Intuitively, in a higher dimensional space, the same number of data points have a better chance of being linearly separable (alternatively, by raising the dimensionality we effectively compute more combinations of the features present in the original representation). The separating hyperplane in the new space, then, corresponds to a non-linear boundary in the original space.

This method can be effective particularly if one's choice of $\phi(\cdot)$ is insightful. However, not only such requisite insight may not be at the researcher's beck and call , but the very process of remapping data, and any subsequent manipulation of them in ϕ-space, can be very expensive, rendering this procedure impractical. Let us see how kernelization can remedy these problems.

Cost of computation In 1964, Aizerman, Braverman, and Rozoner observed that a symmetric positive semi-definite kernel, $k(\cdot, \cdot)$ can be viewed as the inner product of the same function, say, $\phi(\cdot)$, evaluated at two different points, x and y, i.e. $k(x, y) = \langle\phi(x), \phi(y)\rangle$ (the proof of this property is given by Mercer's theorem; Mercer 1909). They further suggested that as long as the learning algorithm only requires the inner products of data points, i.e. $\langle x, y\rangle$, the kernel $k(x, y)$ can be used as a shortcut to first remapping them explicitly through $\phi(\cdot)$, and then computing their inner product. In other words, instead of first computing $x \rightarrow \phi(x), y \rightarrow \phi(y)$ and then $\langle\phi(x), \phi(y)\rangle$, one can compute only the less expensive $k(x, y)$ to the same effect. This shortcut, which came to be known as the kernel trick, made it possible for learning algorithms that upto that point were only effective in linear domains, to successfully handle nonlinear data sets as well, with a reasonable computational overhead. However, it wasn't until 1992 that Boser et al.'s seminal paper on large margin classifiers , also known as Support Vector Machines, made a strong case for the merits of kernelization and introduced it to the mainstream machine learning.

Choice of transformation The foregoing shows how relying on a kernel function can keep the cost of computation under control, however, we still need to figure out what transformation, $\phi(\cdot)$ to use, and also, what kernel $k(\cdot, \cdot)$ corresponds to that particular $\phi(\cdot)$.

Answering the latter question is easy: for a given $\phi(\cdot)$ the corresponding kernel is given by taking its inner product with itself, i.e. $k(x, y) = \langle \phi(x), \phi(y) \rangle$. That the resulting kernel is symmetric follows from the properties of inner product. Positive definiteness is only required to guarantee existence of a corresponding feature map, which in this case would be established independently.

The former question however is not as straightforward, because in general not enough is known about the problem to guide the selection of the right transformation. Consequently, $\phi(\cdot)$, and with it $k(\cdot, \cdot)$ are chosen to be flexible enough to accommodate a wide range of possibilities. In particular, (Cover, 1965) shows that a nonlinearly separable sample set will with high probability become linearly separable after transformation under a dimension-raising map $\phi(\cdot)$. In practice, instead of deciding on $\phi(\cdot)$ and computing the kernel from it, the practitioner decides on an off-the-shelf kernel known to correspond to a dimension-raising $\phi(\cdot)$. Just how high the new dimensionality will be, depends on the particular choice of kernel, which for some cases, e.g. the Gaussian kernel $k(x, y) = Exp(-\gamma||x - y||^2)$, will be infinite (Eigensatz and Pauly, 2006)! Why this is so is beyond the scope of this paper [7], especially since we will not pursue the dimension-raising view of the kernels any further. See table 4.1 for a summary of this discussion.

It is worth emphasizing that in principle, untangling the non-linearly separable data doesn't have to involve raising their dimensionality, and may be achieved via a dimension-preserving (or perhaps even reducing) $\phi(\cdot)$. Therefore, raising the dimensionality is a practical choice that's more convenient than searching for the alternative.

The linear boundary that's obtained in the higher dimensional space, corresponds to a nonlinear boundary in the original space of representation. Naturally, one should worry about the generalizability of the learned criteria: how nonlinear can the decision boundary be before it amounts to over-fitting?

Regularization The high dimensional feature maps induced by kernels make it easy to find the model parameters that fit the training data well. However, on their own, the flexibility that they afford a computational learning system can cripple its performance, since with too good a fit often the training will not generalize well to unseen data. Consider a decision function that wiggles around too often, making it irregular looking. As good as its fit to the training data may be, the decisions that it makes in response to novel situations are questionable. Consequently it is essential that kernel-based learning algorithms employ measures to bound the complexity of their model (cf. VC dimension C_0

[7] The interested reader may observe that expressing the Gaussian kernel in terms of the corresponding $\phi(\cdot)$'s whose inner product would be $k(\cdot,\cdot)$, involves an infinite expansion. For instance, for $x, y \in \mathbb{R}$ we may have $k(x, y) = Exp(-\|x - y\|^2) = Exp(-x^2)Exp(-y^2)Exp(2xy) = Exp(-x^2)Exp(-y^2)\sum_{i=0}^{\infty} \frac{2^i x^i y^i}{i!}$ where the series results from the Taylor expansion of the last term. Therefore, the feature map is $\phi(t) = Exp(-t^2)\sum_{i=0}^{\infty} \sqrt{\frac{2^i}{i!}} t^i$.

$x_1, x_2, ...x_n \in \mathbb{R}^d$	:	Samples are not linearly separable
$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{d_2}, d_2 > d$	:	Dimension-raising map
$\phi(x_1), \phi(x_2), ...\phi(x_n) \in \mathbb{R}^{d_2}$	:	Samples are linearly separable in the new space
$\langle \phi(x_i), \phi(x_j) \rangle$	:	Learning algorithm requires the inner product of the new representations
$k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$	:	Operating in the ϕ-space is expensive; use k instead
$\langle \phi(x_i), \phi(x_j) \rangle = k(x_i, x_j)$	:	Using the less expensive k on the original form of x_i and x_j has the same effect as transforming them with ϕ and taking their inner product

Table 4.1: Summary of the kernel trick from section 4.3.2. In order to apply a linear discriminant algorithm to a sample set that is not linearly separable, one can remap them under a dimension-raising transformation unto a higher dimensional space where they are likely to become linearly separable. Furthermore, to bypass the expense of explicit computation in high-dimensional spaces, a symmetric positive semi-definite kernel can be used in place of the inner product of the samples in the new space.

in section 4.2.2) and keep it as regular as possible. However, just as too much irregularity can degrade performance, too much regularity can be harmful as well. Ideally a learning system needs to be only as irregular (or regular) as the nature of the learning task demands. In machine learning this tension between complexity and simplicity is referred to as bias-variance tradeoff (Geman et al., 1992).

In kernel-based settings this issue is addressed by regularization of the decision boundary (Evgeniou et al., 2000). More specifically, during training, the error that gets minimized includes a term that penalizes the irregularity of the class boundary, e.g. the norm of the derivative of the decision function, $\|f'\|$, which will be smaller for smoother (i.e. more regular) functions.

4.3.3 Kernel as a measure of similarity

So far we have focused on the feature map, $\phi(\cdot)$, and its dimension-raising power for untangling data and making them linearly separable. In fact, were it not for the practical difficulties of working explicitly with $\phi(\cdot)$, there would be no place for the kernel in our discussions. In this section we shift our focus from $\phi(\cdot)$ as a method of increasing dimensionality, to $k(\cdot, \cdot)$ as a measure of similarity.

Recall that according to Mercer's theorem (Mercer, 1909) for any symmetric positive semi-definite kernel there always exist $\phi(\cdot)$ such that $k(x, y) = \langle\phi(x), \phi(y)\rangle$. Therefore, kernels can be viewed as measuring cosine similarity between data points by taking their inner product.[8] However, instead of comparing x and y as they are, k compares a transformed version of them, which for different choices of kernel can be very similar to x and y (e.g. linear kernel: $k(x, y) = \langle x, y\rangle$) or very different (e.g. Gaussian kernel $k(x, y) = Exp(-\gamma||x - y||^2)$). In fact, in this new setting there is no need to invoke the notion of an implicit map $\phi(\cdot)$; the kernel k is any function that assigns a non-negative value to a pair of input points $x_j \in X$ regardless of their order, i.e. $k : X \times X \rightarrow \mathbb{R}$. If the assigned value can serve as a similarity measure (as in the Gaussian kernel where the assigned value $Exp(-\gamma||x - y||^2)$ is a nonlinear form of the Euclidean distance between the inputs $||x - y||$), then that kernel is useful.

In practice, akin to the *Chorus of Prototypes* (Edelman, 1999), a subset of samples are chosen as exemplars, against which the similarity of the remaining samples are measured. The new representation of any to-be-classified point will then be a vector whose j^{th} entry denotes the similarity of that point to the j^{th} exemplar. This new representation can then be used for learning in the usual

[8] For x and $y \in \mathbb{R}^d$, $cos(\theta)$ where θ is the angle between them is given by $\frac{\langle x,y\rangle}{||x||.||y||}$

way. Table 4.2 summarizes these points.

The similarity view of kernels is not a new addition to our repertoire, and has been there all along. Recall that earlier we stated the kernel trick is only useful when the learning algorithm relies on the inner products of the data points. That means that the cosine similarity is built into the dimension raising view too. The similarity view simply targets our interpretation of what the kernel does, and shifts our attention from ϕ as the goal and k as the trick that gets us there, to k itself as the goal. Furthermore, since the kernel trick needs to be accompanied by a learning algorithm that works with inner products, it is often viewed as part of the learning process, evidenced in the strong association between kernel trick and SVM. In the similarity view, on the other hand, the kernel is a means of representing data, somewhat independent of the learning algorithm. The payoff is a representation scheme that is better suited to learning. In fact, under the right circumstances even a simple nearest neighborhood search may suffice for learning from data represented via similarity (cf. section 4.4.3 for examples). A quick query of the literature will show that there now exist a kernelized version of many of the popular algorithms in machine learning.

As an example, let us look at the application of kernels as similarity measures in the Perceptron algorithm. The neurally inspired Perceptron decides on the category of the input by comparing the weighted sum of its components to a threshold: $C(x) = sign(\langle w, x\rangle) = sign(\sum w_j x_j)$[9,10], with w denoting the weight vector. Being a linear combination of the input, Perceptron's decision boundary is a line in the input space and fails to learn the correct category when data are

[9] For simplicity we have dropped the bias term b, the general form of the perceptron decision function is $C(x) = sign(\langle w, x\rangle + b)$

[10] The sign function is defined as $sign(t) = \begin{cases} 1 & t \geqslant 0 \\ -1 & t < 0 \end{cases}$

$X : x_1, x_2, ...x_n \in \mathbb{R}^d$	:	Sample set used in learning
$e_1, e_2, ...e_m \subset X$	:	Subset of the samples chosen as exemplars, e_j
$k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$	:	Appropriate k for measuring similarity
$\mathcal{T} : \mathbb{R}^d \rightarrow \mathbb{R}^m$	:	Re-represent each sample via transformation $\mathcal{T}$. The new dimensionality, m, is decided by the number of exemplars
$\mathcal{T}(x) = (k(x, e_1), k(x, e_2), ...k(x, e_m))$	:	Each sample re-represented as its set of similarities to the exemplars
$\hat{X} : \mathcal{T}(x_1), \mathcal{T}(x_2), ...\mathcal{T}(x_n) \in \mathbb{R}^m$	:	The new representation of samples used in learning

Table 4.2: Summary of the kernel as measure of similarity from section 4.3.3. Instead of a shortcut to high-dimensional computations, the kernel can be viewed as a measure of similarity, yielding a new representation of data that might better serve learning from them. First a subset of samples are chosen as exemplars, and then using the kernel, the similarity of the remaining samples are measured against the exemplars. The new representation of each sample consists of the set of such similarities.

not linearly separable, e.g. the XOR problem (Minsky and Papert, 1969). We should be inclined to kernelize the perceptron in order to fix this shortcoming, however, the kernel trick only works if the learning algorithm requires the inner products only, and never the data points themselves. The original formulation of the perceptron, stated in terms of the weighting of the individual points, does not meet this requirement.

The solution lies in applying the kernel as a measure of similarity. In particular, we select a subset of the training points as exemplars, $e_1, ...e_m$, and measure against them the kernelized similarity of the to-be-classified point, x; the rest is the same as the classical perceptron: $C(x) = sign(\sum_1^m \alpha_j k(x, e_j))$, or in the notation of table 4.2, $C(x) = sign(\langle \alpha, \mathcal{T}(x) \rangle)$. The learning then consists of optimizing α_j, denoting the emphasis we would like to put on the similarity of x to each of the preset exemplars (figure 4.1) (Freund and Schapire, 1999). To see that this new decision rule corresponds to a kernelized version of the linear perceptron, note that the weight vector can be expressed as a linear combination of the exemplars, $w = \sum \alpha_j \phi(e_j)$ – hence, optimizing α has the same effect as optimizing w. Therefore, applying the kernel as a measure of similarity has the same effect as its application in the dimension raising view: $C(x) = sign(\langle w, \phi(x) \rangle) = sign(\langle \sum_j \alpha_j \phi(e_j), \phi(x) \rangle) = sign(\sum \alpha_j \langle \phi(e_j), \phi(x) \rangle) = sign(\sum_j \alpha_j k(e_j, x))$.

Overcomplete representations It is worth pointing out that incorporating nonlinearities of the type discussed thus far can potentially address a range of questions about the response nonlinearities observed in the visual system. Recordings from the early stages of the visual pathway in cats and monkeys indicate certain "non-classical" nonlinearities in how neurons respond to the presence of stimuli (Zetzsche and Rhrbein, 2001). For instance, while stimuli whose neural representation is orthogonal to the weight profile of the receiving unit are expected not to elicit any response, they in fact result in the weakening of that unit's activity. (Olshausen and Field, 1997) argue that this behavior may be explained in light of the sparse and overcomplete coding scheme at work in those areas.

Early in the visual stream (e.g., LGN, V1, V2) the outgoing fibers outnumber the incoming ones, suggesting an overcomplete[11] basis representation scheme, which would result in linear dependency among the firing of different units. Nonlinearly transforming the tuning of the units will counter such dependencies, thus upholding sparseness in their activity (Olshausen and Field, 2004). A thorough investigation of the connections between kernels and overcomplete

[11] An overcomplete basis set is one whose number of bases exceeds its dimensionality. In contrast, a 3×3 identity matrix is not overcomplete in $\mathbb{R}^3$.

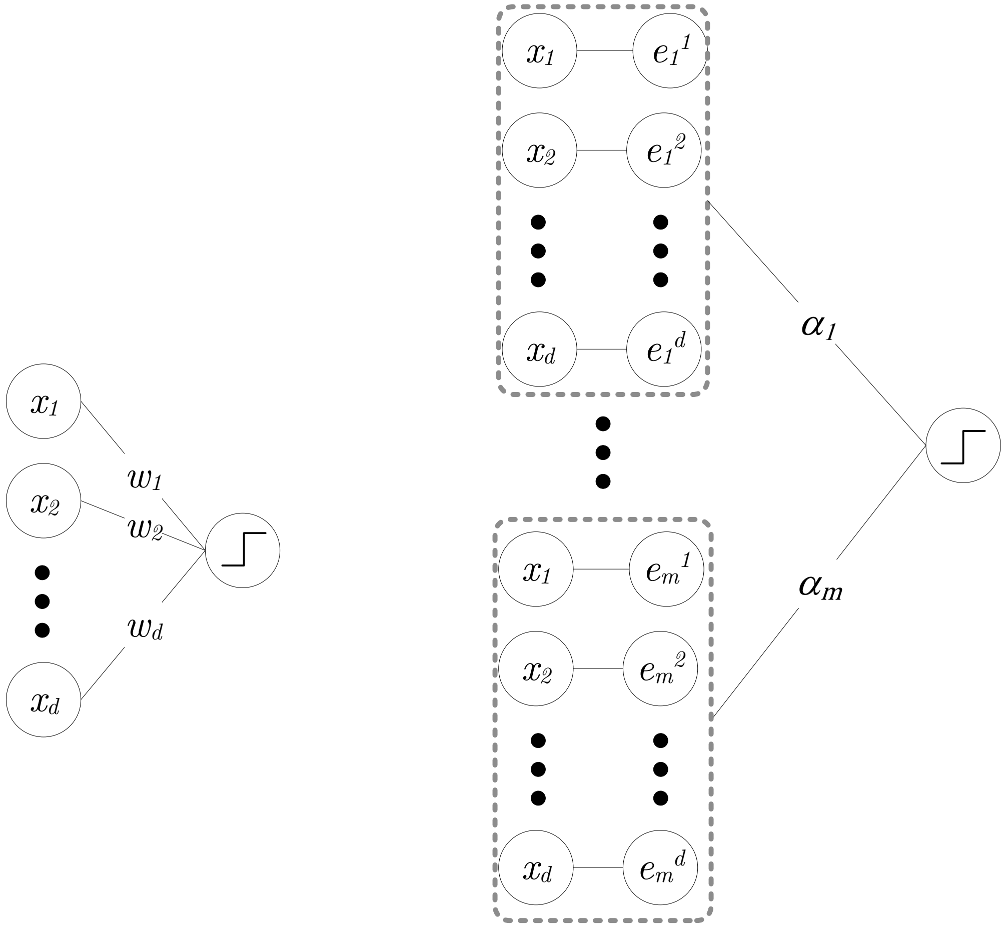


Figure 4.1: *Left*: The classical formulation of the perceptron algorithm can only handle linearly separable data. *Right*: The perceptron algorithm can be modified using kernel as a measure of similarity to become capable of dealing with nonlinearly separable data as well. A subset of data points are chosen as exemplars, and the remaining samples are re-represented as their set of similarities against the exemplars (enveloped by the dashed lines above). α_j, the emphasis put on similarity to exemplar j, plays a similar role to the weights in a non-kernelized perceptron (cf. section 4.3.3 and table 4.2).

representations, however, falls outside the scope of this work.

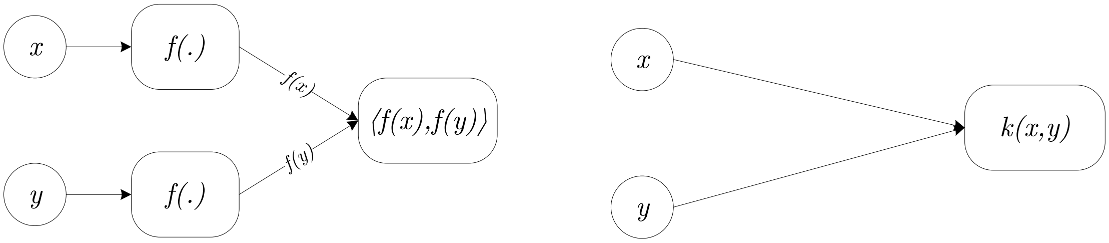


Figure 4.2: *Left*: The rightmost neural processing unit only has access to x and y through $f(\cdot)$. *Right*: If, however, it is only interested in the similarity of x and y, it can use $\langle f(x), f(y)\rangle$ to the same effect as accessing x and y directly (cf. section 4.3.3).

The measurement constraint

Having covered kernels viewed as a shortcut to inner products, and as a measure of similarity, we can now address M_0, the fundamental constraint on measurement.

Suppose x and y are two neural signals each fed into a processing unit whose behavior is captured by $f(\cdot)$. The output of these units is therefore $f(x)$ and $f(y)$ (figure 4.2). A processing unit at the following stage only has access to x and y as modified according to $f(\cdot)$. Therefore, it must rely on computations that do not require explicit knowledge of x and y. However, if the unit is only interested in the similarity of x and y (a reasonable expectations in the nervous system), and the kernel defined by $k(\cdot,\cdot) = \langle f(\cdot), f(\cdot)\rangle$ is a suitable measure of similarity, then this unit can effectively access all the required explicit information about x and y through $k(x, y)$.

In other words, while the similarity notion of the kernel can serve as a means of neural information processing, the implicit feature map may provide a *trick* to circumvent difficulties that arise as a result of limited access to information.

4.3.4 Regarding the dimensionality of kernel solutions

In this section we first reiterate the significance of dimensionality reduction for learning from experience, and then discuss how this issue is addressed in learning systems that rely on kernels.

Dimensionality reduction

Learning from experience requires that the perceptual system consult information obtained for a situation when that situation presents itself later again. However, it is extremely unlikely for two states of the environment to be identical, especially as recorded by the sensory system. Varying conditions such as lighting, pose, clutter, and angle of view, make the retinal imprint of a predator change significantly from one encounter to the next. If the perceptual system is to learn the appropriate response to the presence of a predator (e.g., to flee), it must represent the predator in terms that are insensitive to the details of sensory input. Indeed, for a representation to be useful for future reference, it must admit some abstraction from the measurements performed by the sensory system. Abstractions of this form can usually be obtained by remapping the signal onto a space of smaller dimensionality wherein only certain features of interest from the original signal are retained (Joliffe, 1986; Roweis and Saul, 2000; Hadsell, Chopra, and LeCun, 2006). For instance Principle Component Analysis projects the signal onto the subspace spanned by those dimensions along which the original signal has maximum variance (cf. section 4.2.2).

Computational and economical concerns offer further motivation for dimensionality reduction. Measurements that are recorded by sensory devices are typically of very large dimensions. However, the sample size is often relatively small, giving rise to sparsely distributed data points learning whose relevant statistics can be difficult–i.e. the curse of dimensionality (Bellman, 1961). In addition, high dimensional data burden the system with higher expenses both in terms of storage and performing computations. Reducing the dimensionality of the signal can simultaneously make learning from sparse data more feasible, and lower the system's expenditure.

Do kernels increase or decrease dimensionality?

Kernels' treatment of dimensionality is a somewhat confusing topic. This is in part because the literature is not vocal enough on this issue to guide the uninitiated, and in part because by themselves kernels are indifferent to the dimensionality of their domain of application; whether that dimensionality gets raised, reduced, or left unchanged, is for the most part up to the practitioner. Nonetheless, since in most discussions its usefulness is attributed to the dimension raising power of ϕ, many brain scientists have come to the conclusion that the kernel trick is a method of inflating the dimensionality of data, and thereby, have dismissed it as a irrelevant to the brain; after all, when it comes to the brain, diminishing the dimensionality is a lot more useful than augmenting it (cf. D_0 in 4.2.2 and 4.3.4).

Here we aim to express the relationship between kernels and dimensionality in more explicit and simple (if at times overly so) terms in hopes that other interested researchers may benefit from it.

As it must by now be familiar to the reader, there are two main views on how the kernel does its trick. One view emphasizes the the feature map $\phi(\cdot)$, while the other emphasizes the notion of similarity. Regarding the former view, it is not a necessity that $\phi(\cdot)$ raises the dimension; rather, as discussed under "Choice of transformation" in section 4.3.2, this happens as a matter of convenience, reflecting the designer's imperfect knowledge about the problem. Furthermore, the feature map is *implicit* and never actually computed (which only further complicates this issue). In fact, the sample set never leave their native nominal space, where the eventual solution will also reside. Consequently, the burden of determining the effective dimensionality of the kernelized solution is born by the learning algorithm, not the feature map. This is especially manifest in the imposition of the penalty term to keep the decisions (of the learning algorithm) regular. For instance, in SVM, the effective dimensionality of the decision function is independent of the dimensionality of $\phi(\cdot)$, and lies in the support vectors whose freedom is tightly bound via regularization.

In the similarity view, the dimensionality of the samples does change, but according to the number of exemplars chosen, and, again, independent of the corresponding $\phi(\cdot)$. Consulting table 4.2, you can see that the re-representing transformation $\mathcal{T}$ has dimensionality m, the number of exemplars. The good news is that usually m can take on a value much smaller than d, the dimensionality of the original samples–Section 4.3.4 explores this idea in more detail.

Taken altogether, kernel's treatment of dimensionality does not conflict with the principles of dimensionality and complexity, D_0 and C_0; rather it supports them by confining the dimensionality of the eventual solution, thereby increasing the generalizability of the learned decisions. This is achieved by regularizing the decision function, or, to the same effect, by remapping data using $\mathcal{T}$ which relies on fewer exemplars than the native number of dimensions. This bring us to the next question: How do we choose the exemplars?

Random projections and feature selection

Considering that one mark of a good representation scheme is preservation of relationships (Edelman, 1998, 1999; Shepard, 1987), in machine learning a number of algorithms have been designed whose objective is to reduce the dimensionality of data while preserving their pairwise similarities, often measured in terms of their Euclidean distance. Examples of such algorithms include, PCA, Multidimensional Scaling, Isometric Mapping, and Autoencoders. One property common to most such distance preserving algorithms is that they accomplish their objective by performing carefully designed operations on data (e.g., finding dimensions of maximum variance in PCA, and training to reproduce the input pattern on the output in Autoencoder) which usually end up computationally expensive.

The Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss, 1984) offers a computationally simple and inexpensive way for embedding data into a lower dimensional subspace while preserving their pairwise distances, given that certain conditions hold. More specifically, as long as the number of data points is small relative to their dimensionality, a situation that arises often in perceptual processing, projecting them onto a randomly chosen subspace of much smaller dimensionality will preserve their pairwise distances. Formally, for $x_j \in \mathbb{R}^d$, a linear map $l : \mathbb{R}^d \to \mathbb{R}^m$ with $d \gg m$, and $0 < \varepsilon < 1$ we have

$$(1-\varepsilon)\,\|x_i - x_j\|^2 \leq \|l(x_i) - l(x_j)\|^2 \leq (1+\varepsilon)\,\|x_i - x_j\|^2$$

Following the above considerations, (Balcan et al., 2006) observe that the kernel method can be thought of as a lower dimensional embedding of the data in the following way. Suppose that remapped under ϕ, corresponding to some kernel k, data become linearly separable. Then, following Johnson-Lindenstrauss lemma, projecting the remapped data onto a subspace spanned by randomly chosen vectors r_j should nearly preserve their linear separability. However, the straightforward application of the lemma which is of the form $(\langle r_1, \phi(\cdot)\rangle, \langle r_2, \phi(\cdot)\rangle, ...\langle r_m, \phi(\cdot)\rangle)$[12] would be too expensive to compute, since r_j are of the same dimensionality as ϕ. Instead, one can draw e_1 through e_m from the original data at random to serve as exemplars, and remap any other point x using $\mathcal{T} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ with $d \gg m$ as

$$\mathcal{T}(x) = (k(x, e_1), k(x, e_2), ...k(x, e_m))$$

where $k(\cdot, e_j)$ corresponds to a random projection from ϕ-space along the j^{th} dimension of $\mathcal{T}$, similar to $\langle r_j, \phi(\cdot)\rangle$, but without the explicit computation of ϕ and $\langle r_j, \phi(\cdot)\rangle$. In other words $\mathcal{T}$ provides an inexpensive way to embed x in a lower dimensional subspace while almost preserving its linear separability under ϕ. Formulated this way, k can be thought of as a measure of similarity, and e_j as exemplars, prototypes, landmarks, or features against which k measures the similarity of x (Balcan et al., 2006; Blum, 2006). (Edelman, 1999) and (Anselmi, Leibo, Rosasco, Mutch, Tacchetti, and Poggio, 2014) are two examples of successful

[12] Projection of the vector v onto the subspace spanned by $r_1, ...r_m$ is given by $\mathcal{R} \times v$ where $\mathcal{R}$ is the projection matrix whose j^{th} column is r_j.

application of this method in object recognition.

The small set of randomly selected prototypes in Balcan et al.'s (2006) formulation serves well for a binary classification setting. While the full extent of the significance of feature selection for similarity-based learning systems has not been explored (Pękalska, Duin, and Paclík, 2006), it appears that in a more involved setting with multiple classes and complex feature sets, it may be preferable to *i*) carefully select the exemplars so as to better reflect prior knowledge about the structure of the data *ii*) select them via optimization of certain objective (Klare and Jain, 2012), or *iii*) to increase the number of randomly chosen prototypes, hence heightening the likelihood of covering the would-be-optimal ones.

4.3.5 Kernels and the fundamental constraints

We may now observe that the kernel trick can be used as a basis for an approach that would satisfy all four fundamental constraints listed earlier:

M_0 *The measurement constraint.* By relying on kernels both as a measure of similarity and as an implicit feature map, a neural processing unit can gain explicit access to information that would otherwise be only indirectly accessible (cf. section 4.3.3).

S_0 *The similarity constraint.* A representational unit that needs to compute the similarity of x and y can do so using $k(x, y)$ (cf. section 4.3.3).

D_0 *The dimensionality constraint.* Using the kernel trick, a learning problem can be embedded into a space spanned by the data points, whose effective dimensionality is typically much lower than the nominal dimensionality of the data space (cf. section 4.3.4).

C_0 *The complexity constraint.* By keeping the effective dimensionality of their solutions low, either by regularizing the solution or by relying on few exemplars, kernel-based methods keep complexity under control (cf. section 4.3.4).

4.3.6 A probabilistic angle

Consistent with the prominence of statistical learning in natural and artificial systems, similarity-based methods have received a fair amount of probabilistic treatment. Starting from the basic principles Shepard (1987) derives a universal law for generalization based on similarity stating that the probability of the "consequence" of a novel stimulus being the same as a familiar one decays exponentially in the dissimilarity of the two stimuli, as represented in their "psychological space". This formulation has served as the basis of a number of explicitly or implicitly probabilistic models of classification based on similarity (e.g., Nosofsky 1986; Kruschke 1992). In particular (Tenenbaum and Griffiths, 2001) offer a Bayesian formulation that extends the scope of Shepard's original law to include generalizing from multiple familiar examples.

In the context of nearest neighborhood search, (Kriegel, Kunath, and Renz, 2007) propose a probabilistic formulation where the distance between a query and its neighbors is treated as the probability density of a "hit". Also (Gupta, Gray, and Olshen, 2006) use the relative frequencies of samples that are near neighbors of test point to offer a Bayesian formulation for supervised learning.

Concerning the kernel trick in particular, (Smola, Gretton, Song, and Schölkopf, 2007) propose a way to "embed [probability] distributions in a Hilbert space" by constructing a mapping between the the two and treating each point in the RKHS as the mean of a distribution. This method is further generalized in (Song, Huang, Smola, and Fukumizu, 2009) to include conditional distributions as well, motivating Fukumizu et al. (2011)'s explicitly Bayesian formulation of nonparametric posterior point estimation in RKHS.

4.3.7 Are Gaussian kernels special?

Learning algorithms that make use of the kernel trick prescribe that the kernel be chosen carefully so as to both accommodate the particular set of data at hand and avoid overfitting (see section 4.3.2). In practice, the proper choice of kernel requires that the designer of the application rely on his familiarity and expertise with the problem. For instance, while a polynomial kernel may be all that is needed for certain sets of data, other sets may require a sigmoid or a Gaussian kernel, and even after the right kernel is chosen there still remains the issue of fine tuning its parameters. This task is often referred to as model selection.

The difficulty here is that typically too little is known *a priori* about the problem to guide the choice of kernel. Short of resorting to typically involved methods of automatic model selection (e.g., Howley and Madden 2005) in such cases, especially in visual domains, a popular default choice of kernel is a radial basis function (RBF), often in the form of a Gaussian kernel: $k(x, y) \propto Exp(-||x - y||^2)$, e.g., (Belkin and Niyogi, 2003; Hegde, Sankaranarayanan, and Baraniuk, 2012).

Perhaps not coincidentally, RBF's also seem to be evolution's favorite choice of distance metric in the nervous system. In the primate visual system the strength of response of a cell exhibits an exponential fall off, $Exp(-\gamma||x - y||^p)$, in the distance of the present stimulus, x, from the preferred one, y (Rose, 1979; Daugman, 1980; Kang, Shapley, and Sompolinsky, 2004). This trend is not limited to the visual system and is observed in other cortical regions (Dayan and Abbott, 2001), as well as non-primate species, e.g., (Miller, Jacobs, and Theunissen, 1991; Theunissen, Roddey, Stufflebeam, Clague, and Miller, 1996). Furthermore, following (Shepard, 1987), the behavioral likelihood of generalizing from a familiar stimulus to a novel one is proportional to the negative exponential of their perceived similarity, i.e. their distance in the "psychological space" where they are represented, $P(L_x = l_0|L_y = l_0) \propto Exp(-||x - y||)$ with L_i denoting the label of item i.

A common explanation for this behavior maintains that by being more sensitive to the slope of the tuning curve rather than its magnitude, overlapping graded tuning curves afford the neurons finer discrimination (Snippe and Koenderink, 1992; Butts and Goldman, 2006). Likewise, (Edelman, 1999) makes a similar case for the merits of such receptive fields in object recognition. Here we mention further properties of the Gaussian that may contribute to its frontal role in neural coding.

- The Gaussian tuning curves corresponding to a collection of cells can serve as a basis function set in formation of mappings, for instance between different sensory modalities (Pouget and Sejnowski, 2001). In particular, the basis decomposition of a Gaussian itself results in Gaussians. For instance the Fourier transform of a Gaussian will consist of Gaussian basis functions (e.g. for $f(t) = Exp(-\alpha t^2)$ we have $\mathscr{F}(s) = \sqrt{\pi/\alpha}Exp(-\pi^2 s^2/\alpha)$).

- The Gaussian kernel is self similar, i.e. convolving two Gaussians yields another Gaussian. This property may be helpful in that in a cascade of cells with a Gaussian tuning curve little information is lost to those units that do not have direct access to each other; see M_0.
- The Gaussian can arise from the collective activity of a large population, none of whose individuals are necessarily Gaussian, as stated by Central Limit Theorem.
- The feature map corresponding to a Gaussian kernel is infinite dimensional (Eigensatz and Pauly, 2006), offering more flexibility where little is known about the nonlinearity of data.

Considering that often the cortical tuning curves resemble RBFs, one may ask whether this feat is rooted in our evolutionary history or emerges from the statistics of the environment. More precisely, to what extent is the synaptic weight profile of an RBF-like receptive field coded genetically, as opposed to driven by experience? Research from embryogenesis suggests that certain functional aspects of the nervous system are already in place at birth. For instance (Horton and Hocking, 1996) found that the geniculocortical pathway of macaque monkey shows ocular dominance in prematurely delivered babies. Furthermore, (Wong, 1999) suggests that spontaneous rhythmic bursts of retinal activity in unborn vertebrates can lead to formation of visual receptive fields via Hebbian like mechanisms.

4.4 Issues and ideas related to kernels

In this section, we discuss the main headings under which the relevant material is typically found in the literature. As we shall see, there is considerable overlap among the headings, which underscores the need for a unified framework.

4.4.1 Manifolds and linearization

The dynamics of a an object rotating around an axis can be captured with only one parameter: the angle of rotation. Yet, the measurements recorded from the object by the sensory instruments will not readily present the rest of the perceptual system with this simple dynamics; rather, they provide it with a high dimensional representation–e.g., retinal input–wherein the lower dimensional dynamics are implicit. One of the goals of the perceptual processing is to extract from the sensory measurements the lower dimensional representations that are most pertinent to the task at hand (Edelman and Intrator, 1997). Assuming that the mapping between states of the world and their corresponding percepts is isometric, i.e. small changes in the states of the world are followed by small changes in how they are perceived–note that the intermediate representations need not be, and in practice aren't isometric–then such dynamics of interest will correspond to lower dimensional manifolds embedded in the high dimensional space of their initial measurement (DiCarlo and Cox, 2007). Once the system has learned the appropriate subspace, it can use it in forming invariant representations to facilitate recognition. For instance, in vision, the subspace spanned by the varying viewing conditions of an object can be helpful in building a representation that is invariant to those conditions: as long as the representation lies within the learned subspace, it belongs to the same object. Therefore, one of the goals of perception can be understood in terms of making explicit the more *meaningful* subspaces that are implicit in the sensory signal.

Simultaneously, the task of uncovering such manifolds takes care of another consideration pertinent to perceptual learning. As stated in section 4.2.2, to be effective, the decision strategies learned from experience must generalize well to novel situations. However, while the measurements taken by the sensory devices are high dimensional, informative examples, especially those that figure in vital situations, may be rare (a prey who manages to escape the predator once, may not fare so well upon the next encounter), leading to a sparse set of high dimensional samples with poor generalizability. Relying on the more significant features of the available examples in learning, for instance by uncovering the low dimensional manifold, reduces the effective dimensionality of data and makes them more generalizable. This is akin to Vapnik's observation that when dealing with high dimensional data, one ought to reduce the VC-dimension of the learned decision function to avoid bias and improve generalization (Vapnik, 1999).

The computational techniques developed for the purposes of manifold learning usually rely on some metric that quantifies the similarity of typically high dimensional data points, and use the result to approximate the lower dimensional target manifold (see section 4.4.2 for examples). However the task of these techniques is complicated by the fact that the subspaces corresponding to different stimuli are often irregular and "entangled" (DiCarlo and Cox, 2007; Elgammal and Lee, 2008), untangling which is a challenge similar to learning a nonlinear decision function. Intuitively, finding the boundary between two intertwined manifolds is not unlike finding the boundary separating two sets of data points. Indeed, here too kernelizing the metric used to measure similarity between data points can help to uncover manifolds that will not yield to the straight application of such measures. For instance, while the ordinary application of PCA can only reveal linear trends embedded in data, the kernelized version, kPCA (Schölkopf, Smola, and Müller, 1997), is capable of uncovering nonlinear trends as well. In section 4.4.2 we review application of kernels to a few more manifold learning methods including ISOMAP and Laplacian Eigenmaps.

As in other contexts, application of kernels to manifold learning can also be appreciated in light of their ability to tame expensive computations. For instance, noting that most visual manifold learning methods naively assume isometry to Euclidean distance between the images , (Hegde et al., 2012) suggest the Earth Mover Distance as a better alternative, and apply kernels to break the computational cost of their metric.

4.4.2 Graph methods

In recent years a number of graph theoretic approaches have been proposed whose objective is to discover underlying low dimensional manifolds embedded in the high dimensional space where data points are initially presented. Isometric feature mapping, or Isomap (Tenenbaum, 1998), treats the data points in a small neighborhood as vertices of a weighted graph whose weighted edges correspond to the pairwise Euclidean distances of data points. The geodesic distance of any two points is then the shortest path length between their respective vertices. By applying classical MDS to these geodesic distances, Isomap can reveal underlying nonlinear manifolds in a computationally tractable fashion. Furthermore, the adjacency matrix representing the geodesic distances can be interpreted as a kernel matrix where the weight of the edge connecting vertices i and j corresponds to $k(x_i, x_j)$ (Ham, Lee, Mika, and Schölkopf, 2004), hinting that Isomap may be viewed as a kernel eigenvalue problem, with the caveat that this matrix may not necessarily be positive semi-definite (a property required of kernel matrices).

Similarly, in a method called Laplacian eigenmap, (Belkin and Niyogi, 2003) represent data points on the vertices of a weighted graph whose weighted edges are the Gaussian of the pairwise Euclidean distances of the points, i.e. the weight of the edge connecting points x_i and x_j is $W_{ij} = e^{-\|x_i - x_j\|^2/2\sigma^2}$ (though they call the resultant kernel matrix from this Gaussian a heat kernel, owing to links between their method and the solution to the differential equation of diffusion fields). The target manifold is then computed via eigendecomposition of the graph Laplacian. Furthermore, (Wilson and Zhu, 2008) note that in the spectral analysis of a graph, the Gaussian kernel (heat kernel) representation is preferred because it helps to disambiguate different graphs with the same spectrum. (Sprekeler, 2011) suggests that under certain conditions the Laplacian eigenmap is equivalent to slow feature analysis, a neurally inspired technique that aims to uncover invariant or slowly varying features from high dimensional temporal data that resemble sensory input (Wiskott and Sejnowski, 2002). Finally, (Ham et al., 2004) point out that the above methods, Isometric feature mapping and Laplacian eigenmap, as well as the closely related Locally Linear Embedding (LLE) can in fact be viewed as special cases of the general family of kernel PCA.

4.4.3 Locality-sensitive hashing

Locality Sensitive Hashing (LSH) was originally suggested as an efficient implementation of data retrieval in nearest neighborhood (k-NN) search methods. Though simple and effective (Cover and Hart, 1967), the time complexity of k-NN usually grows intractable as the number of stored data points or their dimensionality increases (Arya, Mount, Netanyahu, Silverman, and Wu, 1998). By relaxing the requirement for an exact match, albeit with a predictably bounded error, LSH offers a k-NN implementation whose cost grows logarithmically in the number of points and their dimensionality (Indyk and Motwani, 1998; Andoni and Indyk, 2008). This efficiency is made possible in part by partitioning the search space into small clusters so that similar entries get binned together, and in part by relying on simple features, or even randomly selected ones (Charikar, 2002), for deciding how to partition the search space.

The clustering aspect of LSH, together with its insensitivity to carefully designed features, makes it a suitable choice for similarity based classification purposes, especially when dealing with large and high dimensional data sets where more conventional methods may be too computationally expensive to be feasible (e.g., Shakhnarovich, Indyk, and Darrell, 2006, and Grauman and Darrell, 2007). Furthermore, equipping the machinery of LSH with the flexibility of kernelized metrics as the measure of similarity has been shown to result in improved performance compared to non-kernelized counterparts such as Euclidean distance (Kulis and Grauman, 2009).

We have previously discussed the merits of Content Addressable Memory (CAM), and the hashing implementation as an instance, for successful evolution of a cognitive agent (Edelman and Shahbazi, 2012; Edelman, 2008). Note that while a conventional hash function is designed to minimize collisions, that is, mapping different entries to the same hash key, in LSH the aim is to maximize collision among entries that are similar enough according to some metric. Nevertheless, LSH can serve as a CAM whereby entries that are similar will receive the same treatment, and as such, it deserves attention as potentially useful means of cognition.

4.5 Similarity and kernels in the brain

4.5.1 Behavioral needs vs. computational means

The very basic needs of an evolving organism–fight, flight, food, reproduction–can take on various forms and shapes in different contexts (table 4.4, right). At a formal level, however, they can typically be stated in terms of few forms of learning and decision making (table 4.3). considering the prevalence of the four fundamental constraints in different domains of cognition, it is not surprising that most forms of learning and decision making can benefit from a kernelized formulation. Table 4.5 summarizes several such cognitive strategies and their kernel algorithm counterparts.

4.5.2 Behavioral findings

Reiterating section 4.2.2, to be useful for survival, neural representations must reflect the similarities of their distal references. At the lower stages of processing the strength of neural responses is a function of the similarity of the current

type of task	*what needs to be done*	*what it takes*
recognition	dealing with novel views of shapes	tolerance to extraneous factors (pose, illumination, etc.)
categorization	dealing with novel instances of known categories	tolerance to within-category differences
open-ended representation	dealing with shapes that differ from familiar categories	representing a novel shape without necessarily categorizing it
structural analysis	reasoning about (i) the arrangement of parts in an object; (ii) the arrangement of objects in a scene	explicit coding of parts & relationships of objects and scenes

Table 4.3: A hierarchy of tasks arising in visual object and scene processing (reprinted from Edelman and Shahbazi, 2012).

stimulus to the preferred one. For instance the firing rate of V1 simple cells decreases as the orientation of the current stimulus moves away from the preferred orientation. At higher stages too, brain imaging techniques suggest that multi-voxel codes well reflect the similarity, both of identity and of structure, of various stimuli presented (Hayworth, Lescroart, and Biederman, 2011; Zhang, Meyers, Bichot, Serre, Poggio, and Desimone, 2011; MacEvoy and Epstein, 2011). What is more, this trend can also be observed in subjects' performance in behavioral tasks (same references).

The flip side of kernel as a measure of similarity is linear separability. Researchers in visual psychophysics have for some years been exploring the effects of linear separability of simple stimuli, defined usually in a low-dimensional

means of survival	*examples*
deciding on an appropriate response to a novel stimulus	"Is this food?" "Is this a dangerous animal?" "Can I outrun this predator?" "How much water do I need for this trip?"
veridical representation	judging the similarity of a red apple to a green apple judging the similarity of a red apple to a red flower
dealing with noise and confounding factors	detecting a lion's roar from a distance in the wind telling apart a dog from a wolf
dealing with ambiguity and missing information	recognizing prey in the fog recognizing an occluded pig by its tail
generalizing learned skills to new tasks	learning to hunt boars can help better hunt deers learning tree climbing can help rock climbing figuring out what a ripe cherry looks like can help figure out what a ripe apricot looks like

Table 4.4: A non-exhaustive list of tasks that can help an animal survive (left column), and examples of situations in which they play out (right column).

parameter space. For instance, Vighneshvel and Arun (2013) used line segments differing only in their orientation as stimuli in a visual search task. In this setting, the task of finding a segment tilted at 0° among 20° and 40° distractors is linearly separable, whereas the task of finding a segment tilted at 20° among 0° and 40° distractors is not. However, natural perceptual categorization problems never reside in such simple spaces. A more realistic approach should vary the layout of stimuli parametrically in some appropriately complex "hidden" space (Cutzu and Edelman, 1996; Op de Beeck, Wagemans, and Vogels, 2001; cf. Blair and Homa, 2001).

means of survival	*possible strategy*	*kernel based ML technique*
deciding on an appropriate response to a novel stimulus	judge similarity to familiar examples	k-NN with kernel metric
	judge similarity to random examples	kernel re-representation $\mathcal{T}(\cdot)$, LSH
	find a decision boundary based on previous examples	SVM, RBF networks
	discover and exploit structure within collected examples	ISOMAP, kPCA, spectral clustering
	quantify output in terms of input	regression, Gaussian processes
veridical representation	preserve pairwise distances	MDS with kernel metric
dealing with noise and confounding factors	allow for variance	regularization
dealing with ambiguity and missing information	use co-occurrent information,	explicit coding of structure, e.g. ChoRD
	top-down processing	generative models - not kernel based (although see section 4.3.6)
generalizing learned skills to new tasks	transfer of learning	hierarchical mixture models - not kernel based

Table 4.5: A non-exhaustive list of visual tasks that can help an animal survive (left column), possible ways these tasks can be undertaken (middle column), and the kernel-based machine learning techniques implementing them (right column).

Although experiments targeted at uncovering the nature of class separability in cortical representations are scarce, there is evidence suggesting that one aspect of class learning can be seen in better linear separation of their representations. For instance, (Miller, Schalk, Hermes, Ojemann, and Rao, 2014) report that electroencephalographic recordings from subcortical areas of 188 epileptic patients can correctly be classified via a linear discriminant classifier. In a parallel project we are collecting imaging data to further investigate the cortical implications of linear separability for learning (cf. section 4.6.2). Behaviorally too, evidence is in favor of linear separability and learning going hand in hand (Medin and Schwanenflugel, 1981; Wattenmaker, Dewey, Murphy, and Medin, 1986; Blair and Homa, 2001). Furthermore, it has been suggested that the metrics entertained by the prototype (Posner and Keele, 1968; Rosch, 1978) and exemplar (Nosofsky, 1988) views of categorization require to operate on representations that respect linear relationships (Blair and Homa, 2001).

4.5.3 The brain angle

Considering that much of the literature on kernels focuses on the issue of linear separability, we may wonder how crucial it is for the representations that the brain uses to be linearly separable. At a first glance it seems that since many of the behavioral needs of an evolving agent rely on computations that resemble problems of machine learning (Table 4.5), linear separation of representations is indispensable in the brain. For example, the formal statement of perceptual categorization is quite similar to that of a myriad classifying algorithms. Indeed as expressed in (DiCarlo and Cox, 2007), (DiCarlo, Zoccolan, and Rust, 2012), and (Pagan et al., 2013) the coarse grains of perceptual information processing in the ventral visual pathway can be cast in terms of untangling the raw sensory measurements in several stages of neural computation. In effect, while the low level subspaces wherein representations of the same class lie are twisted and tangled, higher up the stream they become progressively more amenable to linear decision boundaries.

The chief rationale offered for this picture is invariance: "At higher stages of visual processing, neurons tend to maintain their selectivity for objects across changes in view; this translates to manifolds that are more flat and separated"(DiCarlo et al., 2012). However, at a closer inspection one may notice that the translation from invariance to linear separability is not self-evident and requires appealing to the details of neuron-level mechanics. The linear discriminant approach corresponds closely to one of the main computational building blocks of the brain, the cortical pyramidal cell, whose function is characterized by a linear combination of its inputs, followed by a soft thresholding (Buzsáki, 2010). Therefore, the ability of a neuron (or any hierarchically organized, feedforward, ensemble of them) to tell apart one category from another is only as good as the incoming representations of those categories have been processed by the preceding stages to become linearly separable. Imagine a different universe in which cell responses to input x are characterized by $f(x) = sign(\sum w_j x_j^2)$. In such a universe, a nervous system that expends its resources on untangling the representations with respect to hyperplanes would not fare as well as one that factors in the particular response nonlinearities of its constituent cells.

Linear separability is nonetheless defensible on grounds of complexity. As discussed in section 4.2.2, simpler decision criteria make for more generalizable decisions, in any universe. At this point some clarification is necessary: The linear separation attained by kernels in ϕ-space does not necessarily uphold simplicity, and thereby generalizability. In fact, were it not for the tight grip of the regularizing term, it could easily result in a disastrously overfitted solution.

However, this should not be cause for concern since the untangled neural representations do not reside in the implicit ϕ-space; rather, in a very explicit space comprised of signals that are re-represented by way of certain *flattening* transformation which may or may not involve steps that we can interpret as an implicit, never realized, feature map (cf. section 4.3.4 for more details). Put another way, suppose the nervous system uses the Gaussian kernel to remap the sensory signal residing in S_0 onto a new space, S_1, better suited for the perceptual needs of the organism. Now, while each point in the ϕ-space (corresponding to the Gaussian kernel) stands for an individual point from S_0, each point in S_1 corresponds to the similarity of two points from S_0, measured by the Gaussian kernel. Consequently, one can talk about the decision boundaries in S_0, ϕ, and S_1 spaces which may or may not be hyperplanes (figure 4.3). Extending the example further, representations in S_2, computed as before via a Gaussian kernel but with S_1 as input, would correspond to the similarity of similarities, or meta-similarity. Such higher order measures of similarity, particularly when combined with hierarchic abstraction to reflect the similarity of parts and wholes, have been proven effective in shape (Egozi, Keller, and Guterman, 2010) and string (On and Lee, 2011) matching.

4.6 Conclusions

4.6.1 Summary

An organism that strives to survive and prosper in the wild would benefit from a nervous system that affords it means of making informed decisions in response to the various states of the world. However, such a nervous system is faced with a number of fundamental challenges that it must learn to cope with before

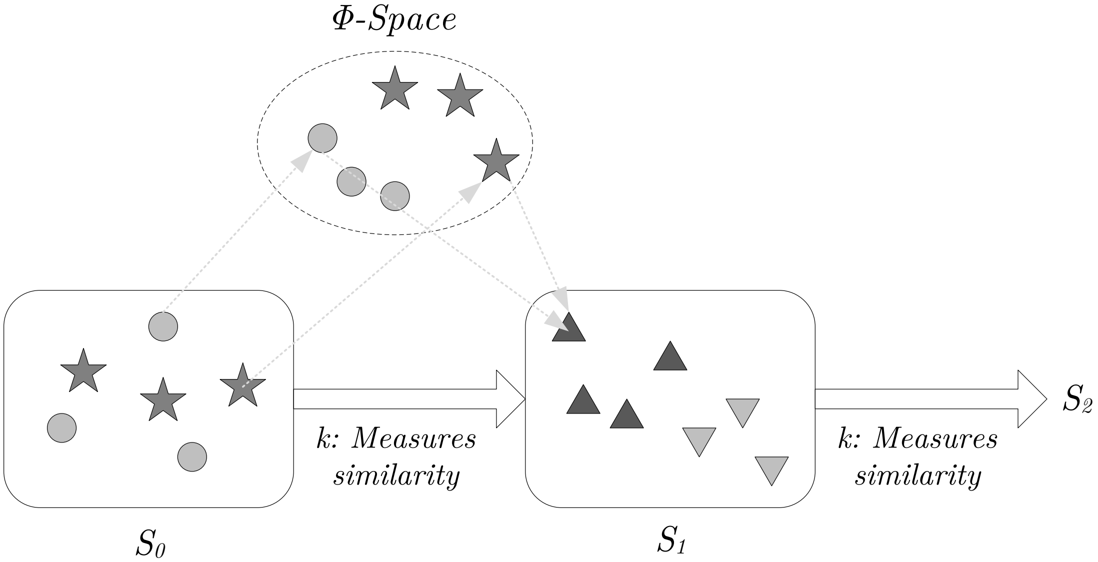


Figure 4.3: Linear separability in neural representations: Raw sensory measurements reside in S_0 where they are typically not linearly separable. Through application of a Gaussian kernel which measures their similarities, they may become linearly separable in the new space S_1. However this linearly separability would be different from the one corresponding to the feature map of the Gaussian, ϕ-space. While each point in ϕ-space corresponds to a sensory measurement, each point in S_1 denotes the similarity of a pair of sensory measurements. Finally, further application of the kernel on S_1 yields S_2 where second-order similarities are represented. See text for details.

it can succeed at its job. We have enumerated four fundamental constraints on the cognitive functioning of a neural information processor that target its essential functioning as well as the economics of its operation–cf. M_0 (section 4.2.1), S_0(section 4.2.2), D_0(section 4.2.2), and C_0(section 4.2.2). With the exception of M_0, each of these challenges have been addressed elsewhere. In this paper we attempted to paint a broad picture based on ideas from Reproducing Kernel Hilbert Spaces, that would relate the various existing approaches to one another.

The kernel trick which was originally conceived as a shortcut to expensive high-dimensional computations can simultaneously address all four constraints. By acting as a measure of similarity (S_0) kernels offer solutions that are of low complexity (C_0) and reside in spaces of lower dimensions (D_0) than the space of training samples. Furthermore, as long as at each stage of processing the required information about earlier stages is limited to comparisons and similarities, relying on kernel-like computations minimizes loss of access to measured information (M_0).

These observations are consistent with findings from a range of biological and behavioral experiments. At the neural level, comparison of input to prototype stimuli and measuring their similarity is among the most common types of neural processing of sensory information. In addition, brain imaging and electrophysiological recordings uphold the notion of low complexity representations afforded by kernel-like computations. Furthermore, behavioral studies on similarity and on linear separability suggest that both factors are involved in subject performance in various forms of learning and decision making.

4.6.2 Future work

As discussed in section 4.3.4, while in simple settings even a randomly chosen set of exemplars may suffice for optimal performance, in real world applications the complexity of data structure (e.g. warped manifolds, skewed distributions, etc. which abound in domains such as image analysis, bioinformatics, and computational genomics) and the demands of the learning task (e.g. multiclass learning) can complicate the story. In such cases randomly chosen features may not always meet the desired performance, and care must be taken to select exemplars that would accommodate the requirements of the learning task. Some research has already been conducted on certain aspects of prototype selection for kernel algorithms (e.g. Laub and Müller, 2004). However, further work is required to find a systematic strategy for selecting prototypes that optimize different performance objectives.

The statement of the fundamental constraint on measurement, M_0, does not entertain a long history in the literature, and as such, our proposed solution to is for the most part theoretic. Further experimentation is needed to explore its neural and behavioral underpinnings and their empirical implications.

From Hume's *Enquiry* (1748) to (Shinkareva, Wang, and Wedell, 2013; Xue, Weng, He, and Li, 2013), the similarity aspect of the framework discussed here has been subject to much theoretical and empirical scrutinizing. However, experimental work on linear separability–the other side of this coin–has only recently started to get the community's attention; e.g., (Miller et al., 2014) for neural, and (Blair and Homa, 2001) for behavioral effects of linear separability. In a separate project we are using fMRI scanning and multivoxel pattern analysis to study the linear separability of higher cortical representations.

The answers of the type we seek are not readily found in the machine learning literature, where the focus is, quite understandably, on improving performance rather than on painting a broad picture that would relate the various existing approaches to one another.

CHAPTER 5

CONCLUDING REMARKS

5.1 Summary and Discussion

The findings of chapters 2 and 3 suggest that not only does learning indeed rely, at least insofar as it concerns vision, on spatial contingencies and structural regularities, but it favors certain such structures over others. This should not surprise us, seeing that vision evolved in an environment abundant with hierarchic organization (which appeared there in the first place because of the evolutionary stability of such organizations). Furthermore, the hierarchic nature of the organization does not have to be strictly apparent to be exploited by the cognitive system. In our experiments, the hierarchicality was only indirectly accessible to subjects through the network of causes that were hidden from them and had to be inferred, perhaps without explicit knowledge. Nonetheless, in the majority of experiments we observed a markedly better performance for the more (covertly) hierarchic conditions.

One of our primary motivations pursued in chapter 4 was to acquaint the brain sciences community with a potentially useful idea that has independently evolved in the discipline of statistical learning. As a result we have in several places sacrificed rigor for ease and clarity in conveying the spirit of the message, which is this:

Representing the sates of the world, and making inferences on them, requires, among other things, quantifying the similarity of those states via metrics that may vary with the demands of the task (though Gaussian-like metrics seem to accommodate a wide range of them). The kernel trick provides an economical means of accomplishing this, while honoring the various constraints that arise in cognitive operation. In particular, computations that involve kernelization, should not, and in practice do not, increase dimensionality; rather, they decrease it in ways that are predictable, and more importantly, controllable by the experimenter.

The prospect of machines that think is starting to appear unsettlingly close (e.g. Stephan Hawking recently expressed his concern over this matter; BBC, 2014). Regardless of the legitimacy of fear over vengeful machines, there is no denying that from the scripted thought processes of (Schank and Abelson, 1977) and SOAR architecture (Laird, Newell, and Rosenbloom, 1987) that marked the beginning stages of cognitive science, our artificially intelligent systems have come a long way. But even steering away from the so called *hard* problems, our learning systems still need to catch up to primate learning on some of the *easy* but fundamental aspects of cognitive function. Structure, briefly discussed in chapter 1 and covered in more detail in chapters 2 and 3, is one example. Representing based on similarities and exemplar/prototype-based decision making, discussed in chapter 4, seems to be in a better standing, already incorporated (implicitly or explicitly) into many learning algorithms.

One major cognitive factor prominent in biological, but for the most part absent in artificial learning, is abstraction. The footprints of abstraction are present in human cognition everywhere from chunking phonemes into words and words into sentences, grouping parts into bigger parts and into objects and scenes, all the way to higher level cognition observed in problem solving and chess playing. Such hierarchic abstraction exists in various disguises in machine learning (e.g. mixture modeling and hierarchic convolutional networks). But abstraction is also a major player in discovering similarities in seemingly disparate domains (e.g. high temporal frequency of sound vs. high spatial frequency of an image). Literature on transfer of learning and Bayesian inference– e.g. hierarchical Dirichlet processes (Teh, Jordan, Beal, and Blei, 2006b) and *machine science* (Evans and Rzhetsky, 2010)–has some pointers on this form of abstraction, but their involvement in machine learning is not as pronounced as in human learning. Perhaps until that comes to change we ought not lose any sleep over the threat of AI powered terminating machines.

5.2 Parting Remarks

> 'What is the meaning of it, Watson?' said Holmes solemnly as he laid down the paper. 'What object is served by this circle of misery and violence and fear? It must tend to some end, or else our universe is ruled by chance, which is unthinkable. But what end? There is the great standing perennial problem to which human reason is as far from an answer as ever.'
>
> –Sherlock Holmes, *The Adventure of the Cardboard Box*

The account of vision and cognition offered in the previous chapters, is, to put charitably, quite technical. Is this how the brain works?

The standard answer to this question would read something along the following lines:

"Marr's *Vision* (1982) [that the first chapter opened with], painted a road map to the proper understanding of mind–brain, by detailing three levels of analysis that any pupil of the brain sciences would have to keep in mind.[1] The short version of Marr's message is this: Any cognition can be upheld by many algorithms realizable via many instantiations. Consequently, to understand mind–brain it is crucial to scrutinize it at many different levels. From molecular and electrophysiological properties of single neurons, collective behavior of neural circuits, propensities of different cortices, functional imaging (where the units of measurement typically contain 600,000 neurons), and electroencephalography (which track the firing of many millions of neurons), to quantifying the behavior of a single human, as well as the dynamics of his interactions with other bodies and the changes that he undergoes in turn, and, of course the more abstract levels which are best captured by formal methodologies, are all essential ingredients in the soup that will be the story of how the mind works."

But let us not bother with the standard answer, and instead turn the question around. For most of its life, physics used to be a qualitative branch of philosophy. It was only around the time when Newton and Leibniz coexisted that we came to realize the far reaches of a formalized physics in explaining the universe. Today we may be comfortable with a deeply computational physics, but the question still remains: Why do the secrets of our universe yield to mathematics?

Atop the threshold of the first Academy, the critical inquisitive methods of whose founder is still at large in its modern namesake, was written in Greek:

"ΑΓΕΩΜΕΤΡΗΤΟΣ ΜΗΔΕΙΣ ΕΙΣΙΤΩ"[2]

[1] These three levels were later revised by Tomaso Poggio (2012).

[2] *Let no one ignorant of geometry enter.*

BIBLIOGRAPHY

Aizerman, A., E. M. Braverman, and L. Rozoner (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control 25*, 821–837.

Andoni, A. and P. Indyk (2008). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM 51*, 117–122.

Anselmi, F., J. Z. Leibo, L. Rosasco, J. Mutch, A. Tacchetti, and T. Poggio (2014). Unsupervised learning of invariant representations with low sample complexity: the magic of sensory cortex or a new framework for machine learning?

Arya, S., D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM) 45*(6), 891–923.

Ashby, F. G. and N. A. Perrin (1988). Toward a unified theory of similarity and recognition. *Psychological Review 95*(1), 124–150.

Baayen, R. H. (2008). *Analyzing linguistic data: A practical introduction to statistics using R*. Cambridge University Press.

Balcan, M.-F., A. Blum, and S. Vempala (2006). Kernels as features: On kernels, margins, and low-dimensional mappings. *Machine Learning 65*, 79–94.

Bates, D. (2005). Fitting linear mixed models in R. *R News 5*, 27–30.

Bateson, P. and R. Hinde (1976). *Growing points in ethology: based on a conference sponsored by St. John's College and King's College, Cambridge*. Cambridge Univ Pr.

BBC (2014). Does ai really threaten the future of the human race? http://www.bbc.com/news/technology-30326384/.

Belkin, M. and P. Niyogi (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation 15*(6), 1373–1396.

Bellman, R. E. (1961). *Adaptive Control Processes*. Princeton, NJ: Princeton University Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Berlin: Springer.

Blair, M. and D. Homa (2001). Expanding the search for a linear separability constraint on category learning. *Memory & Cognition 29*, 1153–1164.

Blum, A. (2006). Random projection, margins, kernels, and feature-selection. In C. Saunders, M. Grobelnik, S. Gunn, and J. Shawe-Taylor (Eds.), *Subspace, Latent Structure and Feature Selection*, Volume Lecture Notes in Computer Science, 3940, pp. 52–68. Springer.

Blumer, A., A. Ehrenfeucht, D. Haussler, and M. Warmuth (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM 36*, 929–965.

Bondy, J. A. and U. S. R. Murty (1976). *Graph theory with applications*, Volume 290. Macmillan London.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152. ACM.

Brainard, D. H. (1997). The psychophysics toolbox. *Spatial vision 10*, 433–436.

Butts, D. A. and M. S. Goldman (2006). Tuning curves, neuronal variability, and sensory coding. *PLoS biology 4*(4), e92.

Buzsáki, G. (2010). Neural syntax: cell assemblies, synapsembles, and readers. *Neuron 68*, 362–385.

Charikar, M. S. (2002). Similarity estimation techniques from rounding algorithms. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pp. 380–388. ACM.

Chater, N., J. B. Tenenbaum, and A. Yuille (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in Cognitive Sciences 10*, 287–291.

Cortes, C. and V. Vapnik (1995). Support-vector networks. *Machine Learning 20*, 273–297.

Cover, T. and P. Hart (1967). Nearest neighbor pattern classification. *IEEE Trans. on Information Theory IT-13*, 21–27.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers 14*, 326–334.

Cutzu, F. and S. Edelman (1996). Faithful representation of similarities among three-dimensional shapes in human vision. *Proceedings of the National Academy of Science 93*, 12046–12050.

Daugman, J. G. (1980). Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research 20*(10), 847–856.

Dawkins, R. (1976). Hierarchical organisation: a candidate principle for ethology.

Dayan, P. and L. Abbott (2001). Theoretical neuroscience: Computational and mathematical modeling of neural systems.

DiCarlo, J. J. and D. D. Cox (2007). Untangling invariant object recognition. *Trends in Cognitive Sciences 11*, 333–341.

DiCarlo, J. J., D. Zoccolan, and N. C. Rust (2012). How does the brain solve visual object recognition? *Neuron 73*(3), 415–434.

Edelman, S. (1998). Representation is representation of similarity. *Behavioral and Brain Sciences 21*, 449–498.

Edelman, S. (1999). *Representation and recognition in vision*. Cambridge, MA: MIT Press.

Edelman, S. (2008). *Computing the mind: how the mind really works*. New York: Oxford University Press.

Edelman, S. and N. Intrator (1997). Learning as extraction of low-dimensional representations. In D. Medin, R. Goldstone, and P. Schyns (Eds.), *Mechanisms of Perceptual Learning*, pp. 353–380. Academic Press.

Edelman, S. and N. Intrator (2002). Models of perceptual learning. In M. Fahle and T. Poggio (Eds.), *Perceptual learning*, pp. 337–353. MIT Press.

Edelman, S. and R. Shahbazi (2012). Renewing the respect for similarity. *Frontiers in Computational Neuroscience 6*, 45.

Egozi, A., Y. Keller, and H. Guterman (2010). Improving shape retrieval by spectral matching and meta similarity. *Image Processing, IEEE Transactions on 19*(5), 1319–1327.

Eigensatz, M. and M. Pauly (2006). *Insights into the Geometry of the Gaussian Kernel and an Application in Geometric Modeling*. Ph. D. thesis, Masters thesis dissertation, Swiss Federal Institute of Technology, Zurich.

Elgammal, A. and C.-S. Lee (2008). The role of manifold learning in human motion analysis. In *Human Motion*, pp. 25–56. Springer.

Epshtein, B., I. Lifshitz, and S. Ullman (2008). Image interpretation by a single bottom-up top-down cycle. *Proceedings of the National Academy of Sciences 105*(38), 14298–14303.

Evans, J. and A. Rzhetsky (2010). Philosophy of science: Machine science. *Science (New York, NY) 329*(5990), 399.

Evgeniou, T., M. Pontil, and T. Poggio (2000). Regularization networks and support vector machines. *Advances in Computational Mathematics 13*, 1–50.

Felleman, D. J. and D. C. Van Essen (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral cortex 1*(1), 1–47.

Fiser, J. and R. N. Aslin (2005). Encoding multielement scenes: Statistical learning of visual feature hierarchies. *Journal of Experimental Psychology 134(4)*, 521–537.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics 7*, 111–132.

Freund, Y. and R. E. Schapire (1999). Large margin classification using the perceptron algorithm. *Machine learning 37*(3), 277–296.

Fukumizu, K., L. Song, and A. Gretton (2011). Kernel bayes' rule. In *NIPS*, pp. 1737–1745.

Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. *Journal of the Institution of Electrical Engineers 93*, 429–441.

Gallian, J. (2010). *Contemporary abstract algebra*. Cengage.

Geman, S., E. Bienenstock, and R. Doursat (1992). Neural networks and the bias/variance dilemma. *Neural Computation 4*, 1–58.

Goldstone, R. L. (1994). The role of similarity in categorization: providing a groundwork. *Cognition 52*, 125–157.

Goodman, N. (1972). *Seven Strictures on Similarity*. Indianapolis: Bobbs Merill.

Grauman, K. and T. Darrell (2007). Pyramid match hashing: Sub-linear time indexing over partial correspondences. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8. IEEE.

Gross, C. G., C. Rocha-Miranda, and D. Bender (1972). Visual properties of neurons in inferotemporal cortex of the macaque. *Journal of neurophysiology*.

Gupta, M. R., R. M. Gray, and R. A. Olshen (2006). Nonparametric supervised learning by linear interpolation with maximum entropy. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 766–781.

Hadsell, R., S. Chopra, and Y. LeCun (2006). Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, Volume 2, pp. 1735–1742. IEEE.

Ham, J., D. D. Lee, S. Mika, and B. Schölkopf (2004). A kernel view of the dimensionality reduction of manifolds. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 47. ACM.

Hastie, T., R. Tibshirani, J. Friedman, and J. Franklin (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer 27*(2), 83–85.

Hayworth, K. J., M. D. Lescroart, and I. Biederman (2011). Neural encoding of relative position. *Journal of Experimental Psychology: Human Perception and Performance 37*, 1032–1050.

Hegde, C., A. C. Sankaranarayanan, and R. G. Baraniuk (2012). Learning manifolds in the wild. *Journal of Machine Learning Research*.

Hirtle, S. C. and J. Jonides (1985). Evidence of hierarchies in cognitive maps. *Memory & cognition 13*(3), 208–217.

Holling, C. S. (2001). Understanding the complexity of economic, ecological, and social systems. *Ecosystems 4*(5), 390–405.

Horton, J. C. and D. R. Hocking (1996). An adult-like pattern of ocular dominance columns in striate cortex of newborn monkeys prior to visual experience. *The Journal of neuroscience 16*(5), 1791–1807.

Howley, T. and M. G. Madden (2005). The genetic kernel support vector machine: Description and evaluation. *Artificial Intelligence Review 24*(3-4), 379–395.

Hubel, D. H. and T. N. Wiesel (1968). Receptive fields and functional architecture of monkey striate cortex. *J. Physiol. London 195*, 215–243.

Hume, D. (1748). *An Enquiry Concerning Human Understanding*. Available online at http://eserver.org/18th/hume-enquiry.html.

Indyk, P. and R. Motwani (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613. ACM.

Intrator, N. and L. N. Cooper (1992). Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks 5*, 3–17.

Jäkel, F., B. Schölkopf, and F. A. Wichmann (2007). A tutorial on kernel methods for categorization. *Journal of Mathematical Psychology 51*, 343–358.

Jäkel, F., B. Schölkopf, and F. A. Wichmann (2008). Generalization and similarity in exemplar models of categorization: Insights from machine learning. *Psychonomic Bulletin & Review 15*, 256–271.

Jäkel, F., B. Schölkopf, and F. A. Wichmann (2009). Does cognitive science need kernels? *Trends in Cognitive Sciences 13*, 381–388.

Johnson, W. B. and J. Lindenstrauss (1984). Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics 26*, 189–206.

Joliffe, I. T. (1986). *Principal Component Analysis*. New York: Springer-Verlag.

Jurica, P., S. Gepshtein, I. Tyukin, and C. van Leeuwen (2013). Sensory optimization by stochastic tuning. *Psychological Review 120*, 798–816.

Kaiser, M. and C. Hilgetag (2010). Optimal hierarchical modular topologies for producing limited sustained activation of neural networks. *Frontiers in neuroinformatics 4*.

Kang, K., R. M. Shapley, and H. Sompolinsky (2004). Information tuning of populations of neurons in primary visual cortex. *The Journal of neuroscience 24*(15), 3726–3735.

Klare, B. and A. Jain (2012). Heterogeneous face recognition using kernel prototype similarities.

Kriegel, H.-P., P. Kunath, and M. Renz (2007). Probabilistic nearest-neighbor query on uncertain objects. In *Advances in databases: concepts, systems and applications*, pp. 337–348. Springer.

Kruschke, J. K. (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review 99*(1), 22–44.

Kulis, B. and K. Grauman (2009). Kernelized locality-sensitive hashing for scalable image search. In *Proc. 12th International Conference on Computer Vision (ICCV)*, pp. 2130–2137.

Laird, J. E., A. Newell, and P. S. Rosenbloom (1987). Soar: An architecture for general intelligence. *Artificial intelligence 33*(1), 1–64.

Laub, J. and K.-R. Müller (2004). Feature discovery in non-metric pairwise data. *Journal of Machine Learning Research 5*, 801–818.

Le, Q. V. (2013). Building high-level features using large scale unsupervised learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 8595–8598. IEEE.

Lehmann, F. (1996). Big posets of participatings and thematic roles. In *Conceptual Structures: Knowledge Representation as Interlingua*, pp. 50–74. Springer.

Locke, J. (1700). *An essay concerning human understanding*.

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, Volume 2, pp. 1150–1157. Ieee.

MacEvoy, S. P. and R. A. Epstein (2011). Constructing scenes from objects in human occipitotemporal cortex. *Nature Neuroscience 14*, 1323–1331.

Marr, D. (1982). *Vision*. San Francisco, CA: W. H. Freeman.

Marr, D. and E. Hildreth (1980). Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences 207*(1167), 187–217.

Maunsell, J. and D. C. van Essen (1983). The connections of the middle temporal visual area (mt) and their relationship to a cortical hierarchy in the macaque monkey. *The Journal of neuroscience 3*(12), 2563–2586.

McCulloch, W. S. (1945). A heterarchy of values determined by the topology of nervous nets. *The bulletin of mathematical biophysics 7*(2), 89–93.

Medin, D. L., R. L. Goldstone, and D. Gentner (1993). Respects for similarity. *Psychological Review 100*, 254–278.

Medin, D. L. and P. J. Schwanenflugel (1981). Linear separability in classification learning. *Journal of Experimental Psychology: Human Learning and Memory 7*(5), 355.

Mercer, J. (1909). Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 415–446.

Miller, J. P., G. A. Jacobs, and F. E. Theunissen (1991). Representation of sensory information in the cricket cercal sensory system. i. response properties of the primary interneurons. *Journal of Neurophysiology 66*(5), 1680–1689.

Miller, K., G. Schalk, D. Hermes, J. G. Ojemann, and R. Rao (2014). 188 decoding the inferior temporal cortex at the speed of perception. *Neurosurgery 61*, 222–222.

Minsky, M. and S. Papert (1969). *Perceptrons*. MIT, Cambridge, Ma.

Modha, D. and R. Singh (2010). Network architecture of the long-distance pathways in the macaque brain. *Proceedings of the National Academy of Sciences 107*(30), 13485–13490.

Murphy, K. et al. (2001). The bayes net toolbox for matlab. *Computing science and statistics 33*(2), 1024–1034.

Neal, R. M. (2000). Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics 9*(2), 249–265.

Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General 115*(1), 39.

Nosofsky, R. M. (1988). Exemplar-based accounts of relations between classification, recognition, and typicality. *Journal of Experimental Psychology: Learning, Memory and Cognition 14*, 700–708.

Olshausen, B. A. and D. J. Field (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research 37*(23), 3311–3325.

Olshausen, B. A. and D. J. Field (2004). What is the other 85% of v1 doing. *Problems in Systems Neuroscience 4*(5), 182–211.

On, B.-W. and I. Lee (2011). Meta similarity. *Applied Intelligence 35*(3), 359–374.

Op de Beeck, H., J. Wagemans, and R. Vogels (2001). Inferotemporal neurons represent low-dimensional configurations of parameterized shapes. *Nature Neuroscience 4*, 1244–1252.

Orbán, G., J. Fiser, R. N. Aslin, and M. Lengyel (2008). Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences 105*(7), 2745–2750.

Orban, G., J. Fiser, R. N. Aslin, and M. Lengyel (2008). Bayesian learning of visual chunks by human observers. *Proceedings of the National Academy of Sciences 105*(7), 2745–2750.

Osgood, C. E. (1949). The similarity paradox in human learning: A resolution. *Psychological Review 56*, 132–143.

Pagan, M., L. S. Urban, M. P. Wohl, and N. C. Rust (2013). Signals in inferotemporal and perirhinal cortex suggest an untangling of visual target information. *Nature Neuroscience*.

Pearl, J. (2000). *Causality: models, reasoning and inference*. Cambridge Univ Press.

Pearl, J. (2001). Causal inference in statistics: A gentle introduction. In *Computing Science and Statistics: Proceedings of Interface '01*, Volume 33. Avaliable online at http://ftp.cs.ucla.edu/pub/stat_ser/R289.pdf.

Pearl, J. (2009). Causal inference in statistics: an overview. *Statistics Surveys 3*, 96–146.

Pękalska, E., R. P. W. Duin, and P. Paclík (2006). Prototype selection for dissimilarity-based classifiers. *Pattern Recognition 39*, 189–208.

Poggio, T. (2012). The levels of understanding framework, revised. *Perception 41*, 1017–1023.

Poggio, T. and S. Ullman (2013). Vision: are models of object recognition catching up with the brain? *Annals of the New York Academy of Sciences*.

Posner, M. I. and S. W. Keele (1968). On the genesis of abstract ideas. *Journal of experimental psychology 77*(3p1), 353.

Pouget, A. and T. J. Sejnowski (2001). Simulating a lesion in a basis function model of spatial representations: comparison with hemineglect. *Psychological review 108*(3), 653.

Resnikoff, H. L. (1989). *The illusion of reality*. New York, NY: Springer.

Rissanen, J. (1987). Minimum description length principle. In S. Kotz and N. L. Johnson (Eds.), *Encyclopedia of Statistic Sciences*, Volume 5, pp. 523–527. J. Wiley and Sons.

Rosch, E. (1978). Principles of categorization. In E. Rosch and B. Lloyd (Eds.), *Cognition and Categorization*, pp. 27–48. Hillsdale, NJ: Erlbaum.

Rose, D. (1979). Mechanisms underlying the receptive field properties of neurons in cat visual cortex. *Vision Research 19*(5), 533–544.

Roweis, S. T. and L. K. Saul (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 2323–2326.

Schank, R. C. and R. P. Abelson (1977). *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. Hillsdale, NJ: L. Erlbaum.

Schölkopf, B., A. Smola, and K.-R. Müller (1997). Kernel principal component analysis. In *Artificial Neural NetworksâĂŤICANN'97*, pp. 583–588. Springer.

Schölkopf, B. and A. J. Smola (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press.

Selfridge, O. G. (1958, November). Pandemonium: a paradigm for learning in Mechanisation of Thought Processes. In *Proceedings of a Symposium Held at the National Physical Laboratory*, London, pp. 513–526. HMSO.

Shahbazi, R., D. J. Field, and S. Edelman (2011). The role of hierarchy in learning to categorize images. In L. Carlson, C. Hölscher, and T. Shipley (Eds.), *Proc. 33rd Cognitive Science Society Conference*, Boston, MA.

Shakhnarovich, G., P. Indyk, and T. Darrell (2006). *Nearest-neighbor methods in learning and vision: theory and practice*.

Shannon, C. (1949). Communication theory of secrecy systems. *Bell system technical journal 28*(4), 656–715.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science 237*, 1317–1323.

Shinkareva, S. V., J. Wang, and D. H. Wedell (2013). Examining similarity structure: multidimensional scaling and related approaches in neuroimaging.

Simon, H. A. (1973). The organization of complex systems. In H. H. Pattee (Ed.), *Hierarchy theory: the challenge of complex systems*, Chapter 1, pp. 1–28. New York: George Braziller.

Simon, H. A. (2002). Near decomposability and the speed of evolution. *Industrial and corporate change 11*(3), 587–599.

Smola, A., A. Gretton, L. Song, and B. Schölkopf (2007). A hilbert space embedding for distributions. In *Algorithmic Learning Theory*, pp. 13–31. Springer.

Snippe, H. P. and J. J. Koenderink (1992). Discrimination thresholds for channel-coded systems. *Biological Cybernetics 66*, 543–551.

Solan, Z., D. Horn, E. Ruppin, and S. Edelman (2005). Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences of the United States of America 102*(33), 11629–11634.

Solomonoff, R. J. (1964). A formal theory of inductive inference, parts A and B. *Information and Control 7*, 1–22, 224–254.

Song, L., J. Huang, A. Smola, and K. Fukumizu (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 961–968. ACM.

Sprekeler, H. (2011). On the relation of slow feature analysis and Laplacian eigenmaps. *Neural Computation 23*, 3287–3302.

Steyvers, M., J. B. Tenenbaum, E. J. Wagenmakers, and B. Blum (2003). Inferring causal networks from observations and interventions. *Cognitive Science 27*, 453–489.

Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei (2006a). Hierarchical dirichlet processes. *Journal of the American Statistical Association 101*(476), 1566–1581.

Teh, Y. W., M. I. Jordan, M. J. Beal, and D. M. Blei (2006b). Hierarchical dirichlet processes. *Journal of the american statistical association 101*(476).

Tenenbaum, J. B. (1998). Mapping a manifold of perceptual observations. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing*, Volume 10, pp. 682–688. MIT Press.

Tenenbaum, J. B. and T. L. Griffiths (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences 24*, 629–641.

Theunissen, F., J. C. Roddey, S. Stufflebeam, H. Clague, and J. Miller (1996). Information theoretic analysis of dynamical encoding by four identified primary sensory interneurons in the cricket cercal system. *Journal of Neurophysiology 75*(4), 1345–1364.

Tversky, A. (1977). Features of similarity. *Psychological Review 84*, 327–352.

Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE Transactions on Neural Networks 10*, 988–999.

Vighneshvel, T. and S. P. Arun (2013). Does linear separability really matter? complex visual search is explained by simple search. *Journal of vision 13*.

von Goldammer, E., J. Paul, and J. Newbury (2003). Heterarchy-hierarchy. two complementary categories of description. *Vordenker Webforum for Innovative Approaches in Science, Economy and Culture*.

Wattenmaker, W. D., G. I. Dewey, T. D. Murphy, and D. L. Medin (1986). Linear separability and concept learning: Context, relational properties, and concept naturalness. *Cognitive Psychology 18*(2), 158–194.

Wilson, R. C. and P. Zhu (2008). A study of graph spectra for comparing graphs and trees. *Pattern Recognition 41*, 2833–2841.

Wiskott, L. and T. J. Sejnowski (2002). Slow feature analysis: Unsupervised learning of invariances. *Neural computation 14*(4), 715–770.

Wong, R. O. (1999). Retinal waves and visual system development. *Annual review of neuroscience 22*(1), 29–47.

Xue, S., X. C. Weng, S. He, and D. W. Li (2013). Similarity representation of pattern-information fMRI. *Chinese Science Bulletin*.

Zetzsche, C. and F. Rhrbein (2001). Nonlinear and extra-classical receptive field properties and the statistics of natural scenes. *Network: Computation in Neural Systems 12*(3), 331–350.

Zhang, Y., E. M. Meyers, N. P. Bichot, T. Serre, T. A. Poggio, and R. Desimone (2011). Object decoding with attention in inferior temporal cortex. *Proceedings of the National Academy of Science 108*, 8850–8855.