SCHEMA SELECTION AND STOCHASTIC INFERENCE IN MODULAR ENVIRONMENTS

Paul Smolensky
Institute for Cognitive Science
University of California, San Diego C-015
La Jolla, CA 92093

ABSTRACT

Given a set of stimuli presenting views of some environment, how can one characterize the natural modules or "objects" that compose the environment? Should a given set of items be encoded as a collection of instances or as a set of rules? Restricted formulations of these questions are addressed by analysis within a new mathematical framework that describes stochastic parallel computation. An algorithm is given for simulating this computation once schemas encoding the modules of the environment have been selected. The concept of *computational temperature* is introduced. As this temperature is lowered, the system appears to display a dramatic tendency to interpret input, even if the evidence for any particular interpretation is very weak.

Introduction

Our sensory systems are capable of representing a vast number of possible stimuli. Our environment presents us with only a small fraction of the possibilities; this selected subset is characterized by many regularities. Our minds encode these regularities, and this gives us some ability to infer the probable current condition of unknown portions of the environment given some limited information about the current state. What kind of regularities exist in the environment, and how should they be encoded?

This paper presents preliminary results of research founded on the hypothesis that in real environments there exist regularities that can be idealized as mathematical structures that are simple enough to be analyzable. Only the simplest kind of regularity is considered here: I will assume that the environment contains modules (objects) that recur exactly, with various states of the environment being comprised of various combinations of these modules. Even this simplest kind of environmental regularity offers interesting learning problems and results. It also serves to introduce a general framework capable of treating more subtle types of regularities. And the problem considered is an important one, for the delineation of modules at one level of conceptual representation is a major step in the construction of higher level representations.

This research was supported by a grant from the System Development Foundation and by contract N00014-79-C-0323, NR 667-437 with the Personnel and Training Research Programs of the Office of Naval Research. During the 1981-82 academic year, support was provided by the Alfred P. Sloan Foundation and Grant PHS MH 14268 to the Center for Human Information Processing from the National Institute of Mental Health.

To analyze the encoding of modularity of the environment, I will proceed in three steps. First, I will describe a general information processing task, completion, for a cognitive system. Then I will describe the entities, schemas, I use for encoding the modules in the environment and discuss how they are used to perform the task. Finally I will offer a criterion for how the encoding of the environment into schemas should be done. The presentation will be informal; more precise statements of definitions and results appear in the appendix.

Completions and Schemas

The task I consider, *completion*, is the inferring of missing information about the current state of the environment. The cognitive system is given a partial description of that state as input, and must produce as output a completion of that description.

The entities used to represent the environmental modules are called *schemas*. A given schema represents the hypothesis that a given module is present in the current state of the environment. When the system is given a partial description of the current state of the environment, the schemas look at the information to see if they fit it; those that do become *active*. When inference from the given information to the missing information is possible, it is because some of the schemas represent modules that incorporate both given and unknown information. Such a schema being active, *i.e.* the belief that the module is present in the current environmental state, permits inferences about the missing information pertaining to the module. (Thus if the given information about a word is **ALG###THM**, the schema for the module *algorithm* is activated, and inferences about the missing letters are possible.)

There is a problem here with the sequence of decision-making. How can the schemas decide if they fit the current situation when that situation is only partially described? It would appear that to really assess the relevance of a schema, the system would first have to fill in the missing information. But to decide on the missing portions, the system first needs to formulate beliefs concerning which modules are present in the current state. I call this the *schema/inference decision problem*; what is desired is a way of circumventing it that is general and extremely simple to describe mathematically.

Harmony and Computational Temperature

Before discussing the algorithm used to tackle this problem, let us first try to characterize what outcome we would like the algorithm to produce. The approach I am pursuing assumes that *the best inference about the missing information is the one that best satisfies the most hypotheses (schemas).* That is, consider all possi-

ble responses of the system to an input (a partially specified environmental state). Each such response involves (a) a decision about exactly which schemas are active, and (b) a decision about how to specify all the missing information. Given such a response, I propose a measure of the *internal consistency* that computes the degree to which the hypotheses represented by the active schemas are satisfied by the input and output. I call this measure the *harmony function H*. Responses characterized by greater internal consistency - greater harmony - are "better." (The definition of H is simple, but requires the formalism given in the appendix.)

Given that better responses are characterized by greater harmony, the obvious thing to do is hill-climb in harmony. However this method is apt to get stuck at local maxima that are not global maxima. We therefore temporarily relax our desire to go directly for the "best" response. Instead we consider a *stochastic* system that gives different responses with different probabilities: the better the response, the more likely the system is to give it. The degree of spread in the probability distribution is denoted T; for high values of T, the distribution is widely spread, with the better responses being only slightly more likely than less good ones; for low values of T, the best response is much more likely than the others. Thus T measures the "randomness" in the system; I call it the *computational temperature*. When we want only good responses, we must achieve low temperature.

A general stochastic algorithm can be derived that realizes this probabilistic response; it provides a method for computer simulation. A parallel relaxation method is used to resolve the schema/inference decision problem in a way that involves no sophisticated control.

The variables of the system are the activations of all the schemas (1 = active, 0 = inactive) and the bits of missing information. The system starts with (a) all schemas inactive, and (b) completely random guesses for all bits of missing information. Then randomly a schema or a bit of missing information is selected as the variable to be inspected; it will be now be assigned a (possibly new) value of 1 or 0. Using the current guesses for all the other variables, the harmony function is evaluated for the two possible states of the selected variable. These two numbers, $H(1)$ and $H(0)$, measure the overall consistency for the two cases where the selected variable is assigned 1 and 0; they can be computed because tentative guesses have been made for all the schema activations and missing information. Next a random choice of these 1,0 states is made, with the probability of the choices 1 and 0 having ratio $\frac{e^{H(1)/T}}{e^{H(0)/T}}$. (The reason for the exponential function is given in the appendix.)

This random process - pick a schema or bit of missing information; evaluate the two harmonies; pick a state - is iterated. In the theoretical limit that the process continues indefinitely, it can be proved that the probability that the system gives any response is proportional to $e^{H/T}$, where H is the harmony of that response. This probability distribution satisfies the qualitative description of response probabilities we set out to realize.

Cooling the System

In this algorithm, each schema activation or bit of missing information is determined randomly; most likely the value with higher harmony is chosen, but sometimes not. Thus most of the time the changes raise the harmony, but not always. The higher the temperature, the more random are the decisions, that is, the more often the changes go "downhill." Thus this algorithm, unlike strict hill-climbing, does *not* get stuck at local maxima. However, there is a tradeoff. The higher the temperature, the faster the system escapes local maxima by going downhill; but the higher the temperature, the more random is the motion and the more of the time the system spends in states of low harmony. Eventually, to be quite sure the system's response has high harmony, the temperature must be low.

As the computation proceeds, the optimization point of this tradeoff shifts. Initially, the guesses for the missing information are completely random, so the information on which schemas are determining their relevance is unreliable. It is therefore desireable to have considerable randomness in the decision making, *i.e.* high temperature. Even at high temperature, however, the system is more likely to occupy states of higher harmony than lower, so the guesses become more reliable than their completely random start. At this point it makes sense for the schemas to be somewhat less random in their activity decisions, so the temperature should be lowered a bit. This causes the system to spend more of its time in states of higher harmony, justifying a further decrease in temperature. And so the temperature should be gradually lowered to achieve the desired final condition of low temperature. (A central concern of future work is analysis of how to regulate the cooling of the system.)

As the computation proceeds and the temperature drops, the system's initially rough and scattered response becomes progressively more accurate and consistent. This is just the kind of computation typical in people and just the kind needed in any large parallel system, where each subsystem needs a constant stream of input from the others.

Cognitive Crystalization

As computation proceeds, does accuracy increase slowly and steadily, or does the system undergo sudden and dramatic changes in behavior, as do physical systems when they are cooled past critical temperatures marking phase transitions? This question has been addressed both through computer simulation and analytic approximation of a two-choice decision. The system has two schemas representing conflicting interpretations of the environment. The approximate theory allows computation of the probabilities of various completions given an input that partly describes the environment. It is useful to ask, *what is the completion of a completely ambiguous input?* For high temperatures, as one might expect, the completions form random mixtures of the two interpretations; the system does not choose either interpretation. However, *below a certain "freezing temperature," the system adopts one of the interpretations.* Each interpretation is equally likely to be selected. The computer simulation approximates this behavior; below the freezing point, it flips back and forth between the two interpretations, occupying each for a long time.

Slowly vacillating interpretation of genuinely ambiguous input is a familiar but not particularly important feature of human cognition. What is significant here is that *even when the input provides no help whatever in selecting an interpretation, the system eventually (when cooled sufficiently) abandons meaningless mixtures of interpretations and adopts some coherent interpretation.* A robust tendency to form coherent interpretations is important both for modelling human cognition and for building intelligent machines. The above analysis suggests that in processing typical inputs, which

are at most partially ambiguous, as processing continues and the temperature drops, the system wanders randomly through an ever-narrowing range of approximate solutions until some time when the system freezes into an answer.

Schema Selection

Having discussed how schemas are used to do completions, it is time to consider what set of schemas ought to be used to represent the regularities in a given environment. Suppose the system experiences a set of states of the environment and from these it must choose its schemas. Call these states the *training set*. Since the schemas are used to try to construct high-harmony responses, a reasonable criterion would seem to be: *the best schemas are those that permit the greatest total harmony for responses to the training set.*

I call this the *training harmony criterion*. I will not here discuss an algorithm for finding the best schemas. Instead, I shall present some elementary but non-obvious implications of this very simple criterion. To explore these implications, we create various idealized environments displaying interesting modularity. Choosing a training set from within this environment, we see whether the training harmony criterion allows the system to *induce the modularity from the training set*, by choosing schemas that encode the modularity.

Perceptual Grouping

We perceive scenes not as wholes, nor as vast collections of visual features, but as collections of objects. Is there some general characterization of what is natural about the particular levels of grouping that form these "objects"? This question can be addressed at a simple but abstract level by considering an environment of strings of four letters in some fixed font. In this idealized environment, the modules ("objects") are letter tokens, and in various states of the environment they recur exactly: each letter always appears in exactly the same form. The location of each of the four letters is absolutely fixed; I call the location of the first letter the "first slot," and so forth. The environment consists of all combinations of four letters.

Now consider a computer given a subset of these four-letter strings as training; call these the *training words*. The image the computer gets of each training word is just a string of bits, each bit representing whether some portion of the image is on or off. The machine does not know that "bit 42" and "bit 67" represent adjacent places; all bits are spatially meaningless. Could the machine possibly induce from the training that certain bits "go together" in determining the "first letter", say? (These, we know, represent the first slot.) Is there some sense in which schemas for the letter *A* in the first slot, and so on, are natural encodings of this environment?

As an obvious alternative, for example, the system could simply create one schema for each training word. Or it could create a schema for each bit in the image. These are the two extreme cases of maximally big and small schemas; the letter schemas fall somewhere in between. Which of these three cases is best? *The training harmony criterion implies that letter schemas are best*, provided the training set and number of bits per letter are not too small.

This result can be abstracted from the reading context in which it was presented for expository convenience; the mathematical result does not depend upon the interpretation we place upon the modules with which it deals. Thus the result can be characterized more abstractly: *Natural schemas encoding the modules of an environment are inducible by the training harmony criterion*, provided the modules recur exactly. This investigation must now be extended to cases in which the recurrence of the modules is in some sense approximate.

Rules *vs.* Instances

When should experience be encoded as a list of instances and when as a collection of rules? To address this issue we consider two environments that are special subsets of the four-letter environment considered above:

Environment R

FAMB	*VEMB*	*SIMB*	*ZOMB*
FAND	*VEND*	*SIND*	*ZOND*
FARP	*VERP*	*SIRP*	*ZORP*
FALT	*VELT*	*SILT*	*ZOLT*

Environment I

FARB	*VAMP*	*SALT*	*ZAND*
FENP	*VELB*	*SEMD*	*ZERT*
FIMT	*VIRD*	*SINB*	*ZILP*
FOLD	*VONT*	*SORP*	*ZOMB*

In the highly regular environment *R*, there are strict rules such as "*F* is always followed by *A*"; in the irregular environment *I*, no such rules exist. Note that here, schemas for the "rules" of environment *R* are just *digraph* schemas: *FA--*, *VE--*, ..., *--RP*, *--LT*; schemas for "instances" are whole-word schemas.

One might hope that a criterion for schema selection would dictate that environment *R* be encoded in digraph schemas representing the rules while environment *I* be encoded in word schemas representing instances. *The training harmony criterion implies that for the regular environment, digraph schemas are better than word schemas; for the irregular environment, it is the reverse.* (In each case the entire environment is taken as the training set.)

Higher Level Analyses

The framework described here is capable of addressing more sophisticated learning issues. In particular, it is well-suited to analyzing the construction of higher-level representations and considering, for example, the value of hierarchical organization (which is not put into the system, but may come out in appropriate environments.) In addition to addressing issues of schema selection at the more perceptual levels considered here, the framework can be employed at higher conceptual levels. The selection of temporal scripts, for example, can be considered, by taking the environment to be a collection of temporally extended episodes. The simulation method described here for systems with given schemas can also be applied at higher levels; it is being explored, for example, for use in text comprehension.

ACKNOWLEDGEMENTS

I am indebted to George Mandler and Donald Norman for the opportunity to pursue this work. Geoffrey Hinton, James McClelland, David Rumelhart, and the members of the UCSD

Parallel Distributed Processing research group have contributed enormously to the perspective on cognition taken in this work. I am grateful to Francis Crick, Mary Ann McCrary, Michael Mozer and Donald Norman for extremely helpful comments on the paper. Above all, warm thanks go to Douglas Hofstadter for his continual encouragement; the influence of his ideas pervades this research.

APPENDIX: The Formal Framework of Harmony Theory

In the following general discussion, the specifics of the four-letter grouping environment discussed in the text are presented in parentheses.

The possible beliefs B of the cognitive system about the current state of the environment forms a space **B**. This space is assumed to have a set **p** of binary coordinates p; every belief B is defined by a set of bits $B(p)$ in $\{+1,-1\}$, one for each $p \in \mathbf{p}$. (Each p represents a distinct pixel. $B(p)$ is +1 if the system believes p is on, -1 if off. This belief can come from inference or input.) An *input* I to the system is a set of binary values $I(p)$ for *some* of the $p \in \mathbf{p}$; an *output* is a set of binary values $O(p)$ for the remaining p. Together, I and O form a complete belief state B, a *completion of* I.

A *schema* S is defined by a value $S(p) \in \{+1,-1,0\}$ for each $p \in \mathbf{p}$. (If pixel p does not lie in the first slot, the schema S for "the letter A in the first slot" has $S(p) = 0$. If p does lie in the first slot, then $S(p)$ is ± 1 according to whether the pixel is on or off.) If for some particular p, $S(p) \neq 0$, then that p is an *argument* of S; the number of arguments of S is denoted $|S|$. (For a word schema, every p is an argument.)

Schemas are used by the system to infer likely states of the environment. For a given environment, some schemas should be absent, others present, with possibly varying relative strengths corresponding to varying likelihood in the environment. A *knowledge base* for the system is a function σ that defines the relative strengths of all possible schemas: $\sigma(S) \geq 0$ and $\sum_S \sigma(S) = 1$. (The knowledge bases relevant to the text have all $\sigma(S) = 0$ except for the schemas in some set **S** - like letters - for which the strengths are all equal. These strengths, then, are all $1/\#\mathbf{S}$, the inverse of the number of schemas in the set.)

A *response* of the system to an input I is a pair (A,B), where B is a completion of I, and A defines the *schema activations*: $A(S) \in \{0,1\}$ for each schema S.

A *harmony function* H is a function that assigns a real number $H_\sigma(A,B)$ to a response (A,B), given a knowledge base σ, and obeys certain properties to be discussed elsewhere. A particular exemplar is

$$H_\sigma(A,B) = \sum_S \sigma(S)\, A(S) \left[\sum_p B(p) S(p) - \kappa |S|\right]$$

Here κ is a constant in the interval [0,1]; it regulates what proportion ϵ of a schema's arguments must disagree with the beliefs in order for the harmony resulting from activating that schema to be negative: $\epsilon = \frac{1}{2}(1-\kappa)$. In the following we assume ϵ to be small and positive. (The terms without κ make H simply the sum over all active schemas of the strength of the schema times the number of bits of belief (pixels) that are consistent with the schema minus the number that are inconsistent. Then each active schema incurs a cost proportional to its number of arguments, where κ is the constant of proportionality.)

The probability of any response (A,B) is a monotonically increasing function f of its harmony $H(A,B)$. f is constrained by the following observations. If p_1 and p_2 are not connected - even indirectly - through the knowledge base σ, then inferences about p_1 should be *statistically independent* of those about p_2. In that case, H is the *sum* of the harmony contributed by three sets of schemas: those connected to p_1, those connected to p_2, and those connected to neither. The desired statistical independence requires that the individual probabilities of responses for p_1 and p_2 *multiply* together. Thus f must be a function that takes *additive* harmonies into *multiplicative* probabilities. The only continuous functions that do this are the exponential functions, a class that can be parametrized by a single parameter T; thus

$$\text{prob}(A,B) = n\, e^{H(A,B)/T}$$

The normalization constant n is chosen so that the probabilities for all possible responses add up to one. T must be positive in order that the probability increase with H. As T approaches zero, the function f approaches the discontinuous function that assigns equal nonzero probability to maximal harmony responses and zero probability to all others.

Considerations similar to those of the previous paragraph lead in thermal physics to an isomorphic formula (the Boltzmann law) relating the probability of a physical state to its energy. The randomness parameter there is ordinary temperature, and I therefore call T the *computational temperature* of the cognitive system.

The probability of a completion B is simply the probability of all possible responses (A,B) that combine the beliefs B with schema activations A: $\text{prob}(B) = n \sum_A e^{H(A,B)/T}$.

Monte carlo analyses of systems in thermal physics have proven quite successful (Binder, 1976). Starting from the exponential probability distribution given above, the stochastic process described in the text can be derived (as in Smolensky, 1981). The above formula for H leads to a simple form for the stochastic decision algorithm of the text that supports the following interpretation. The variables can be represented as a network of "p nodes" each carrying value $O(p)$, "S nodes" each carrying value $A(S)$, and undirected links from each p to each S carrying label $\sigma(S) \cdot S(p)$ (links labelled zero can be omitted). The nodes represent stochastic processors running in parallel, continually transmitting their values over the links and asynchronously setting their values, each using only the labels on the links attached to it and the values from other nodes transmitted over those links. This representation makes contact with the neurally-inspired "connectivist" approach to cognition and parallel computation (Hinton and Anderson, 1981; McClelland and Rumelhart, 1981; Rumelhart and McClelland, 1982). Independently of the development of harmony theory, Hinton and Sejnowski (1983) developed a closely related approach to stochastic parallel networks following (Hopfield, 1982) and (Kirkpatrick, Gelatt, and Vecchi, 1983). From a nonconnectionist artificial intelligence perspective, Hofstadter (1983) is pursuing a related approach to perceptual grouping; his ideas have been inspirational for my work (Hofstadter, 1979).

An *environment* for the cognitive system is a probability distribution on **B**. (All patterns of on/off pixels corresponding to sequences of four letters are equally probable; all other patterns have probability zero.) A *training set* **T** from this environment is a sample of points T drawn from the distribution. (Each T is a

training word.) A *response* **A** of the system to the set **T** is a specification of schema activations $A(T)$ for each T. The *training harmony* of such a response is $\mathbf{H}_\sigma(\mathbf{A},\mathbf{T}) = \sum_T H_\sigma(A(T),T)$. The maximum of $\mathbf{H}_\sigma(\mathbf{A},\mathbf{T})$ over all responses **A** is $\mathbf{H}_\sigma(\mathbf{T})$, the *training harmony permitted by the knowledge base* σ.

Each of the sets of schemas considered in the text (letters, digraphs, ...) *tile* the training set **T**, in that each T agrees exactly with schemas that have nonoverlapping arguments and that together cover all of T. All results cited in the text follow from this elementary calculation: *If the knowledge base* σ consists of a set of schemas **S** *that tile* **T**, *then* $\mathbf{H}_\sigma(\mathbf{T}) = \text{const.}/\#\mathbf{S}$ *where* const. *is a constant for a given* **T**. Thus given a single training set **T** and two tilings of **T** with sets of schemas, the set with fewer schemas permits greater training harmony, and is preferred by the training harmony criterion. If the number of letters in the alphabet a is smaller than the number of pixels per letter, letters are a better encoding than pixels; if the number of training words exceeds $4a$ then letters are better than words. The number of word schemas needed in the restricted environments R and I is 16; the number of digraphs needed for R is 8 while for I it is 96.

The calculation cited in the previous paragraph is quite simple. Recall that if the proportion of a schema's arguments that disagree with the beliefs exceeds ϵ, then the harmony resulting from activating that schema is negative. Thus if ϵ is chosen small enough (which we assume), then for any given training word T, only those schemas that match exactly can contribute positive harmony. In the response $A(T)$ that maximizes the harmony, therefore, only exactly matching schemas will be active; the others will be inactive, contributing zero harmony. Since the schemas **S** tile **T**, for each pixel p in T there is exactly one active schema with p as an argument, and the value $B(p)$ of the pixel is consistent with that schema, so $B(p)S(p) = 1$. Thus

$$\sum_S A(S) \sum_p \left[B(p)S(p) - \kappa \mid S(p) \mid \right] = \#\mathbf{p}(1-\kappa) = \#\mathbf{p} \cdot 2\epsilon$$

Because $\sigma(S)$ is a constant for all $S \in \mathbf{S}$, the harmony $H_\sigma(A(T),T)$ is simply the previous expression times $\sigma(S) = 1/\#\mathbf{S}$, or $2\epsilon\#\mathbf{p}/\#\mathbf{S}$. Since this quantity is identical for all training words T, summing over all T in the training set **T** just multiplies this by the size of **T**, giving $2\epsilon\#\mathbf{p}\#\mathbf{T}/\#\mathbf{S}$ as the training harmony permitted by σ:

$$\mathbf{H}_\sigma(\mathbf{T}) = \text{const.}/\#\mathbf{S}$$

where const. $= 2\epsilon\#\mathbf{p}\#\mathbf{T}$, which is constant for a given **T**.

REFERENCES

Binder, K. "Monte Carlo Investigations of Phase Transitions and Critical Phenomena." In Domb, C. and M. S. Green (Eds.), *Phase Transitions and Critical Phenomena*, vol. 5b. New York: Academic Press, 1976.

Hopfield, J. J., "Neural networks and physical systems with emergent collective computational abilities." *Proc. National Academy of Sciences USA* 79 (1982) 2554-2558.

Hinton, G. E. and J. A. Anderson (Eds.). *Parallel Models of Associative Memory*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1981.

Hinton, G. E. and T. J. Sejnowski, "Analyzing Cooperative Computation" in *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*. Rochester, New York, May, 1983.

Hinton, G. E. and T. J. Sejnowski, "Optimal Perceptual Inference," to appear in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. Washington, D.C., June, 1983.

Hofstadter, D. R. *Godel, Escher, Bach: an Eternal Golden Braid*. New York: Basic Books, 1979. Chapters X, XVI and XIX.

Hofstadter, D. R., "The Architecture of Jumbo," to appear in *Proc. Machine Learning Workshop*, Illinois, June, 1983.

Kirkpatrick, S., C. D. Gelatt and M. P. Vecchi, "Optimization by Simulated Annealing." *Science* 220:4598 (1983) 671-680.

McClelland, J. L. and D. E. Rumelhart, "An interactive activation model of context effects in letter perception, Part 1: An account of the basic findings." *Psychological Review* 88 (1981) 375-407.

Rumelhart, D. E. and J. L. McClelland, "An interactive activation model of context effects in letter perception, Part 2: The contextual enhancement effect and some tests and extensions of the model." *Psychological Review* 89 (1982) 60-94.

Smolensky, P. *Lattice Renormalization of* ϕ^4 *Theory*. Doctoral dissertation, Physics Dept., Indiana University, 1981.