An Algebraic Approach to Similarity and Categorization

Mehdi Dastani
Department of Computational Linguistics
University of Amsterdam
(mehdi@ai.let.uva.nl)
Bipin Indurkhya
Department of Computer Science
Tokyo University of Agriculture and Technology
(bipin@cc.tuat.ac.jp)

Abstract

We extend Leeuwenberg's Structural Information Theory (SIT) to an algebraic framework, where domain-dependent operators are allowed to become part of the perceptual gestalts. Using the SIT notion of 'information load', we provide a structural notion of similarity and categorization. We introduce two complexity measures, namely descriptor complexity and member complexity, which drive categorization to opposite extremes, and propose a simple additive function to find an optimum balance between the two. We remark on how context effect in similarity and categorization can be modeled in our approach. Finally, we examine some implications of our work for the empirical research on categorization and similarity.

Introduction and Background

In Leeuwenberg (1971), a coding system for perceptual patterns, called *Structural Information Theory* [1] [SIT, henceforth], is introduced. In this system, alternative perceptual structures are constructed by means of a set of operators: namely, *iteration, symmetry* and *alternation.* A notion of *information load* is defined on these perceptual structures, and it is claimed that the perceptual gestalts correspond to the structures carrying the minimum information load. (See also Van der Helm and Leeuwenberg 1991.)

In our previous work, we have proposed an algebraic formalization for modeling the process of change of representation and contextual interpretation underlying certain analogies (Indurkhya 1991). More recently, we have incorporated SIT's notion of information load into our formalization, thereby producing what might be considered an algebraic version of SIT, and modeled perceptual proportional analogies in it (Dastani, Indurkhya and Scha 1997). We consider this algebraic version to be more powerful than SIT in that it allows any class of operators to be used for structural decomposition; in particular, domain dependent regularities are easily represented in it, and can influence the gestalts. We have applied this approach to model context effect — meaning how in different contexts different structures of the same object are seen to have a different information load, and are consequently selected as gestalts (Indurkhya and Dastani 1997) — and are in the process of extending it to develop a representation system for visual patterns, and to formalize the process of visualization of information (Dastani 1997).

In this paper, we apply our algebraic framework to the problems of similarity and categorization. In particular, we are interested in exploring the implications of our approach to how an object is given different interpretations in different contexts, and how similarity criteria are changed depending on the accompanying objects. We also extend our formalization to model categorization: given a number of objects, what is the most natural way to group them into categories. We examine some consequences of our formalization and how they relate to existing empirical research.

This paper is organized as follows. In the next section we briefly outline SIT and our algebraic extension of it. Then we will make some brief comments on how similarity and context effect are modeled in our formalization. Following that we will extend our framework to model categorization and relate this model to empirical findings. Finally, we will present the conclusions of this paper and discuss future research directions.

Structural Information Theory

The idea underlying SIT is that perceptual descriptions of patterns are based on certain regularities. These regularities are defined in terms of the identity relation among the constituents of patterns. For example, the regularity of the pattern *abab* is defined in terms of the identity of the two substrings *ab*. In order to cover these regularities and determine perceptual descriptions of patterns, SIT defines a number of rules that are used to parse a pattern into descriptions. These rules are based on three operators as follows:

Iteration rule, e.g. : $aaa = 3 \star (a)$
Symmetry rule, e.g. : $abcddbca = S[(a)(bc)(d), ()]$
Alternation rule, e.g. : $abacdae =$
$<(a)> / <(b)(cd)(e)>$

[1] In fact, the coding system is designed for discrete one-dimensional patterns like character strings. We are now developing an alternative coding system for two-dimensional visual patterns.

An expression resulting from applications of these operators [henceforth, ISA operators] to a pattern provides a description of that pattern. These descriptions have the predicate-argument form. The arguments in the descriptions are called *chunks* which indicate the constituents or parts of the pattern under that description. Note that the recursive application of these rules to a pattern will result in a hierarchical structure which reflects further decompositions of chunks into smaller chunks. Moreover, the parentheses used in the description of a pattern indicate the *units* of elements to which the ISA operators are applied. In fact, only the symmetry operator is sensitive to the units of elements, for it reflects the units of elements without reflecting the internal structure of united elements, as illustrated above.

A description of a pattern represents a specific gestalt of that pattern. Different rules may be applicable to a pattern such that different descriptions, representing different gestalts of the pattern, result. In order to disambiguate this set of alternative gestalts and determine the preferred perceptual gestalt, an empirically validated measure is introduced which assigns a complexity value to each gestalt. This complexity measure is called *Information Load* and is computed by adding the number of occurrences of individual elements in the gestalt description (without including the ISA operators) to the number of occurrences of units that contain more than one element. For example, the information load of $S[(a)(bc),()]$ would be $3+1=4$ because it has three elements (a, b and c) and there is one unit (namely, bc) with more than one element.

The gestalt of a pattern which has the lowest information load is claimed to reflect the preferred perceptual description of that pattern. For example, the pattern $aaaa$ has, among others, $S[(a), 2\star(a)]$ and $4\star(a)$ as two possible gestalt descriptions. The information loads for these gestalts are 2 and 1, respectively, and so the second description, namely $4\star(a)$, represents the preferred perceptual description of the pattern $aaaa$. The idea that the simplest gestalt reflects the preferred perceptual description is called the *simplicity principle*. The underlying claim here is that the simplest description of a pattern is the one that employs the smallest number of primitive elements. In this way, SIT formalizes one aspect of the Prägnanz notion of gestalt psychology, viz. descriptive economy; *the information load of what is actually perceived is lower than the information load of what could have been perceived alternatively.*

An Algebraic version of SIT

One can consider the gestalt of patterns as the terms of an algebra $< D, F >$, called *SIT-algebra*, for which the domain of elements, D, is the set of all one-dimensional string patterns and F is the set of ISA operators. The set of one-dimensional string patterns is defined as follows:

Definition 1 *The domain D of one-dimensional string patterns is specified as the class of all finite strings generated by the grammar $G = (V, T, P)$ where:*
$V = \{S, A\}$ is the set of non-terminal symbols,
$T = \{a, \ldots, z\}$ is the set of terminal symbols,
S is the special non-terminal for the start symbol, and
P is the following set of rewrite rules:

$$\{\ S \rightarrow A,\quad A \rightarrow (A),\quad A \rightarrow AA,\quad A \rightarrow a \mid \ldots \mid z\ \}.$$

Notice that perceptual structures of patterns covered by SIT are based on regularities which are in terms of identities between constituents of the patterns. For example, the regularity of the pattern $abab$ is in term of the identity relation between the first ab and the second ab, which is reflected by its perceptual description $2\star(ab)$. However, in certain domains we may be interested in regularities which are in terms of, not only the identity relation, but other domain dependent relations as well. For example, we may want to cover regularities of strings in terms of relations like 'successor', 'predecessor', etc. In this way, the string abc is considered as a regular pattern because c is the 'successor' of b, which is in turn the 'successor' of a. In fact, this is a regularity which is based on the domain dependent successorship relation. These kinds of regularities will be called *domain-dependent regularities.*

In SIT, the perceptual structures of patterns are in terms of ISA operators. There are four operators that are explicitly mentioned in SIT, called Symmetry, Iteration, right and left Alternation, and two operators that are not explicitly mentioned. We call these two operators as the *Concatenation* operator and the *Unit* operator. The concatenation operator concatenates elements while the unit operator determines the units of elements to which the ISA operators apply. These ISA operators can be modified such that they generate descriptions based not only on the identity relation but also on other domain-dependent relations. We concentrate on domain-dependent regularities that can be covered by iteration and alternation operators. At this point we choose to exclude the symmetry operator because it can be considered as a special case of the modified iteration operator, i.e. two times iteration with respect to the reflection relation.

One way to redefine the iteration operator such that it covers regularities based upon domain-dependent relations is by allowing domain-dependent relations as arguments to the iteration operator. In order to redefine the alternation operator, we follow its definition from the original version of SIT where the alternation operator is defined to cover structures which consist of two interspersed structures with one of them described by an iteration operator. Thus, once we allow the modified iteration operator to be based on the basis of domain-dependent relations, the alternation operators can also be based on domain-dependent relations. We define two different alteration operators, called the right and the left alternations, which differ from each other by the sequential order of the two interspersed structures.

Definition 2 *Let $X, Y_1, \ldots, Y_n$ be string patterns. We write $f^m(X)$ to indicate an embedded application (m times) of an operator f to an element X, i.e. $f^m(X) = f(\ldots(f(X))\ldots)$; m times. The algebraic (modified) ISA operators are then characterized as follows:*

$Iter(X, m, f) \rightarrow\ X\ f^1(X) \ldots f^{m-1}(X)$

$R\_Alt(X, f, < Y_1, \ldots, Y_n >) \rightarrow$
$\quad X\ Y_1\ f^1(X)\ Y_2 \ldots f^n(X)\ Y_n$

$L\_Alt(X, f, < Y_1, \ldots, Y_n >) \rightarrow$
$\quad Y_1\ X\ Y_2\ f^1(X) \ldots Y_n\ f^n(X)$

$Con_n(X_1, \ldots, X_n) \rightarrow X_1 \ldots X_n,$

$Unit_n(X_1 \ldots X_n) \rightarrow (X_1 \ldots X_n).$

For example, the descriptions for the strings *abc*, *ccbbaa*, *abcddbca* and *afbkct* are then as follows:

$abc = Iter(a, 3, succ)$
$ccbbaa = Iter(Iter(c, 2, Id), 3, pred)$
$abcddbca = Iter(Con(a, Unit(Con(b, c)), d), 2, reflect)$
$afbkct = R\_Alt(a, succ, < f, k, t >)$

Thus, according to the modified version of ISA operators the string *abc* has an iteration structure such that *abc* is considered as one single chunk. Note that the modified version of iteration operator does not specify any regularity in the pattern *kfu* and so its perceptual chunking consists of three subchunks, i.e. *k*, *f*, and *u*. The way an element is built up out of others is given by a structural description of that element; this is called a *term* of the algebra.

Definition 3 *The class of structural descriptions over an algebra $< D, F >$, denoted by $T_F(D)$, can be recursively defined as follows:*
1) For all $t \in D$, $t \in T_F(D)$, and
2) If $f \in F(n)$ and $t_1, \ldots, t_n \in T_F(D)$, then
$\quad f(t_1, \ldots, t_n) \in T_F(D)$.

In order to allow abstract description of patterns, instead of a description of one particular pattern, one can introduce variables standing for terms. This means extending the SIT-algebra $< D, F >$ with a denumerable infinite set of variables X, i.e. $< D \cup X, F >$. The recursive definition of $T_F(D)$, is then extended by the following clause:

3) If x is a variable, then $x \in T_F(D)$.

An element from the set of patterns is associated with each variable-free term of the SIT-algebra. The assignment of patterns to variable-free terms are defined as follows:

Definition 4 *The assignment of patterns to terms can be defined by means of an evaluation function, E. For any variable-free term, $t \in T_F(D)$, the evaluation function, E, can be defined as follows:*
1) If $t \in D$, then $E(t) = t$, and
2) if t is a term of the form $f(t_1, \ldots, t_n)$ with $f \in F(n)$, then
$\quad E(t) = f(E(t_1), \ldots, E(t_n))$

Based on the evaluation function, the extensional equality (which is not the structural identity) of terms can be defined. Two terms are extensionally equal if and only if they both get evaluated to the same pattern. The terms of the SIT-algebra correspond to the SIT descriptions, which means that the SIT terms describe gestalts of patterns. Consequently, extensionally identical terms constitute different gestalts of the same pattern.

Thus, SIT can be described in terms of the SIT-algebra, $< D, F >$, together with a complexity function, C, which assigns a natural number to each term of $< D, F >$, i.e. $C(t) \in \mathbf{N}$, where t is a term of $< D, F >$. The minimum principle states that the preferable gestalt of a pattern from D is the term, among all extensionally equal terms evaluating to D, which has the lowest complexity value.

Similarity and Context Effect in SIT-Algebra

To model the notion of similarity, and how it changes in different contexts, we need to introduce the concept of *subalgebra.* Essentially, a subalgebra of an algebra $< A, F >$ is another algebra $< B, G >$ such that B is a subset of A, $G(n)$ is a subset of $F(n)$ for all n, and whenever $b_1, \ldots, b_n \in B$ and $g \in G(n)$, then $g(b_1, \ldots, b_n) \in B$. Now the subalgebras of a SIT-algebra can be used to represent contexts. The general idea is that it seems quite unreasonable to require that a cognitive agent consider *all* possible domain-dependent structural operators and choose a gestalt with the absolute minimum. Instead, in any given context, there may be only a few operators that are active, and the cognitive agent would focus on only those gestalts that can be expressed in terms of these operators, and choose a minimum information load gestalt. Thus, when one is perceiving an object, and already has formed a gestalt for it, the structural operators occurring in this gestalt would affect the gestalt of the objects in its spatial and temporal vicinity. (See also Indurkhya and Dastani 1997.)

We have applied this framework to model how gestalts of different terms in a proportional analogy relation 'A is to B as C is to D' emerge so as to minimize the overall information load (Indurkhya 1991, Dastani 1996): the model is applied to the domain of geometric figures, and Hofstadter's (1995) character string domain. Here we limit ourselves to making a few general observations about this approach.

The first thing to note is that the individual minimum information load gestalts do not necessarily lead to the minimum overall information load. Thus, when a pattern is placed in the context of different patterns, different gestalts of the pattern are selected according to the minimum information load principle.

The second thing to point out is that the goal of the cognitive activity constrains how the information load of the overall situation is determined, and what role the gestalts of the constituents play in it. For proportional analogy (of the form 'A is to B as C is to D'), we refer to this extra constraint as 'projectibility condition', and it essentially says that there be a SIT-algebra, say A_1, for terms A and B, another SIT-algebra, say A_2 for terms C and D, and there be an isomorphism between A_1 and A_2. (The isomorphic condition may seem to be too strong, but it really is not. As we are dealing with subalgebras, it is easy to discard the unnecessary structural operators from both algebras, and focus only on mapped operators. Further, one-to-many and many-to-one mappings can also be allowed by indexing the operators appropriately.) We will see another example of this goal-dependent constraint in the next section when we extend our model to include categorization.

Finally, we note that once we start incorporating context in our model as representation algebras, there are other factors that begin to contribute towards the information load. For instance, complexity of context algebras themselves becomes a factor in determining overall information load. To this, we could also add the complexity of the mapping (isomorphic relation between the two algebras). At this point, we are still working on incorporating all these factors in our model.

Categorization in SIT-Algebra

We now proceed to extend the SIT-algebra approach to address the categorization problem. We use the character-string domain of Hofstadter (1995) to illustrate our points. Other domains like the domain of geometrical figures can be covered as well by including their domain dependent relations in the SIT-algebra (Dastani 1996).

SIT-algebra provides us with structural descriptions of character strings, which can be used to define categories, and an information load measure, which can be used to determine the distance between categories and establish an optimal categorization scheme. This idea is illustrated by the following example. Consider the character strings *aabbcc*, *stts*, *pqr*, *mmwwmm*. Applying SIT to each isolated character string results in its set of possible structural descriptions. For instance, some descriptions for *aabbcc* will be:

$Iter(Iter(a,2,id),3,succ)$,
$Iter(X,3,succ)$,
$Iter(X,N,\omega)$,
$Con(Iter(a,2,id),Y,Iter(c,2,id))$,
X

for *stts* will be:
$Con(s,Iter(t,2,id),s)$,
$Iter(Con(s,t),2,reflection)$,
$Iter(X,N,\omega)$,
X

for *pqr* will be:
$Iter(p,3,succ)$,
$Iter(X,3,succ)$,
$Iter(X,N,\omega)$,
$Con(X,q,Z)$,
X

and for *mmwwmm* will be:
$Con(Iter(m,2,id),Iter(w,2,id),Iter(m,2,id))$,
$Iter(Con(Iter(m,2,id),w),2,reflection)$,
$Iter(X,N,\omega)$,
$Con(X,Y)$,
X

Note that some of these descriptions do not contain variables and thus describe only one element. In contrast, the abstract descriptions containing variables describe sets of elements. For example, $Iter(Iter(a,2,id),3,succ)$ describes only the character string *aabbcc* while $Iter(X,3,succ)$ can be seen to describe both *aabbcc* and *pqr* by assigning $Iter(a,2,id)$ and p to the variable X, respectively. Note that $Iter(X,N,\omega)$, $Con(X,Y)$, or X describe all the character strings above.

Structural descriptions may thus be considered as identifiers of categories of elements. In the case of descriptions without variables the identified categories contain only one element and in the case of abstract descriptions the identified categories may contain more than one element. The abstract descriptions describe the structural similarity of the elements of the categories that they identify.

Definition 5 *A category C of elements (character strings) is defined as a pair $C = < CD, A >$ consisting of a category descriptor CD (algebraic SIT term) and a set A of assignment functions. Each assignment function, f_j, assigns a SIT term to each variable which occurs in the category descriptor, i.e.* $f_j : VAR \rightarrow SIT\_term$. *The categorized elements of the category C result from applying the assignment functions from A to the variables occurring in the descriptor CD.*

A category scheme CS for a set of elements is then a set of categories which can describe the whole set of elements.

Of course, given a set of character strings there are many categorization schemes possible. Following the definition of category scheme, the above example of character strings results, among others, in the following three category schemes:

$CS_1 = \{C_1, C_2\}$
where

$C_1 =< Iter(X, 3, succ),$
$\{ \{(X, Iter(a, 2, id))\} ,$
$\{(X, p)\} \} >$

and

$C_2 =< Iter(X, 2, reflection),$
$\{ \{(X, Con(Iter(m, 2, id), w))\} ,$
$\{(X, Con(s, t))\} \} >.$

$CS_2 = \{C_1\}$
where

$C_1 =< Con(X, Y, Z),$
$\{ \{(X, Iter(a, 2, id)), (Y, Iter(b, 2, id)),$
$(Z, Iter(c, 2, id))\},$
$\{(X, p), (Y, q), (Z, r)\},$
$\{(X, Iter(m, 2, id)), (Y, Iter(w, 2, id)),$
$(Z, Iter(m, 2, id))\},$
$\{(X, s), (Y, Iter(t, 2, id)), (Z, s)\} \} >$

$CS_3 = \{C_1, C_2\}$
where

$C_1 =< Iter(X, N, \omega),$
$\{ \{(X, Iter(a, 2, id)), (N, 3), (\omega, succ)\},$
$\{(X, Con(s, t)), (N, 2), (\omega, reflection)\} \} >$

and

$C_2 =< Con(X, Y),$
$\{ \{(X, p), (Y, Con(q, r))\},$
$\{(X, Con(Iter(m, 2, id), w)),$
$(Y, Con(w, Iter(m, 2, id)))\} \} >.$

Note that according to SIT, the minimum description of an element describes the preferred (cognitive and perceptual) structure of that element viewed in isolation. However, in categorization, the elements interact with each other such that the preferred descriptions of categorized elements are affected by this interaction. This means that the preferred description of an element viewed in the context of other elements may be different than the preferred description of that element viewed in isolation. Consequently, the optimal categories are not necessarily defined in terms of the preferred descriptions of their elements viewed in isolation. Therefore, we should consider the descriptions of elements in the context of each other in order to determine the optimal categories. In this way, the categorization is considered as a context within which elements should be interpreted. One way to accomplish this is to require that the descriptions of all elements must result in a minimum overall information load.

In order to determine the optimal (or perceptually preferred) categorization schemes, among all possible categorization schemes, we measure two quantities of each categorization scheme which are defined in terms of information load. The first quantity is the complexity of the category descriptors which we will call *descriptor complexity*. The second quantity is the complexity of the terms provided by assignment functions that together with the category descriptor specify uniquely the categorized elements. We will call this quantity *member complexity*.

The descriptor complexity of a category is the information load of the term which describes the similarity among all members of that category. By summing over the descriptor complexities of all categories, we can measure the descriptor complexity of a category scheme.

Definition 6 *Let* $C_i =< CD_i, A_i >$ *be a category with the descriptor complexity* $COMP_D$*, i.e.*
$COMP_D(C_i) = I(CD_i)$*, where* $I(CD_i)$ *is the information load of the category descriptor* CD_i*.*
Then, given a category scheme $CS = \{C_1, \ldots, C_n\}$*, its descriptor complexity* $COMP_D(CS)$ *is defined as follows:*
$COMP_D(CS) = \sum_{i=1}^{n} \ COMP_D(C_i)$*.*

Assuming that the information load of a SIT expression is influenced by all parameters like the primitive elements, all kinds of variables, the domain dependent relations and the number indicating the iteration times, the descriptor complexity of the categorization scheme CS_1 for the above example, with descriptors $Iter(X, 3, succ)$ and $Iter(X, 2, reflection)$, may be computed as:

$COMP_D(CS_1) = COMP_D(C_1) + COMP_D(C_2) =$
$I(Iter(X, 3, succ)) + I(Iter(X, 2, reflection)) =$
$3 + 3 = 6.$

The second thing to measure is the member complexity of category schemes. As noticed, the member complexity is the complexity of the SIT terms, provided by the assignment functions, which are needed, beyond the category descriptor, to specify the categorized elements uniquely. So, for *aabbcc*, the SIT term for the category descriptor $Iter(X, 3, succ)$ would be $Iter(a, 2, id)$. In order to measure the member complexity of a category, we measure the complexities of their corresponding assignment functions which are defined in terms of the information loads of their range elements. Then, the member complexity over all the categories becomes the member complexity of the categorization scheme.

Definition 7 *Let* $C_i =< CD_i, A_i >$ *be a category with* A_i *containing* m *assignment functions*
$f_j^i : VAR \rightarrow \{t_1^i, \ldots, t_p^i\}$*, where* $t_1^i, \ldots, t_p^i$ *are SIT terms.*
The complexity of each assignment function f_j^i *is defined as the sum of the information loads* I *of its range elements, i.e.*
$COMP(f_j^i) = \sum_{k=1}^{p} I(t_k^i)$*.*
Then, the member complexity $COMP_M$ *for a category* C_i *is considered as the sum of the complexities of its assignment functions, i.e.*
$COMP_M(C_i) = \sum_{j=1}^{m} COMP(f_j^i)$*.*

Finally, given a category scheme $CS = \{C_1, \ldots, C_n\}$, *the member complexity of the category scheme* $COMP_M(CS)$ *is defined as follows:*
$COMP_M(CS) = \sum_{i=1}^{n} COMP_M(C_i)$.

These two quantitative measures drive the categorization to opposite extremes. So, if we group all the elements into one category, then the descriptor complexity of the categorization scheme is minimal, while its member complexity is maximal. On the other hand, if we consider each object in a category by itself, then the member complexity due to categorization scheme is minimal, while the descriptor complexity of the categorization scheme is maximal.

Having the descriptor complexity and the member complexity, the goal is to find a balance between these two extremes, i.e. a way to combine the two measures like adding them together, or multiplying the two. Following the SIT idea of minimum information load and considering the preferred structures of elements in the context of categorization of elements, where the interaction of elements play a role in the computation of their descriptions, these two complexity measures should be added such that the preferred categorization scheme can be defined as the one with the minimum overall information load. For example, taking again the above example of three categorization schemes and considering the minimum overall information load (adding descriptor complexity and member complexity) as the way to select the optimum category scheme, the first category scheme CS_1 becomes the preferred or optimum categorization scheme.

This algebraic approach to the categorization problem can be extended to formalize the intuitive idea that categorization is in principle hierarchical in nature. In this way, a set of elements can be divided into a number of categories, each of which contains a subset of elements. Each of these categories which contains more than one element can also be subdivided into a number of subcategories. This sub-categorization process can be continued until all categories contain only one element. In order to extend our approach to formalize this intuition, we redefine categories recursively as follows:

Definition 8

- *If* t *is a primitive term, then* t *is a category,*
- *If* SC *is a non-empty set of categories and* CD *is an abstract description containing variables, then* $< CD, CS >$ *is a category.*

Note that this recursive definition implies that a category scheme is the same thing as a category.

In order to determine the optimal categorization scheme, we apply both the descriptor and the member complexities to all possible categorization schemes and decide the optimal categorization as the one with the minimum overall information load.

Relevant Empirical Research

We would now like to make a few remarks in connecting our model with some of the empirical research in this area. First of all, we draw attention to the fact that most of the empirical work on similarity and categorization uses some version or another of multidimensional feature space, with little emphasis on structural aspects. (See, for example, Goldstone, Medin and Halberstadt 1997, Pevtzow and Goldstone 1994, Rodet and Schyns 1994, Stins and Van Leeuwen 1993.) Some studies that have explored the structural aspects of similarity and categorization have also focused on the alignment and matching of structures — whether relations or attributes are used as a basis of alignment, etc. (See, for example, Goldstone 1994). As far as are aware, there has been little effort at studying how complex structures are encoded, how they carry information, and how they affect similarity judgments and categorization.

Now in a general way, we believe that the context effects found by Tversky (1977), and more recently elaborated in the experiments of Goldstone, Medin and Halberstadt (1997), could be explained as shifts from one subalgebra to another. Moreover, how categorization affects object decomposition (Pevtzow and Goldtone 1994), and how the order in which categories are learnt affect their encoding (Rodet and Schyns 1994), can be modeled as a kind of hysteresis effect whereby the representation algebras from the previous context influence the subsequent cognitive tasks. For example, Rodet and Scyns (1994) found that the subjects who learned category XY before learning category X did not possess the feature to distinguish Y, but the subjects who learned category X before learning XY did possess this feature; even though both groups of subjects performed equally well in identifying exemplars of both categories X and XY.

Similarly, an explanation can perhaps be put forward for the context effects observed by Stins and Van Leeuwen (1993). They found that when the subjects were given an integrative task (determining whole-part relationship), they could be influenced by the prime, but not when they were given a non-integrative task (comparing an internal angle of two figures). An integrative task triggers structural decomposition, in which representation algebras play a crucial role (which can be influenced by primes).

More importantly, however, our approach does have some non-trivial implications, even for experimental data that is based on multi-dimensional feature space (with essentially no structure). For example, in Experiment 3 by Goldstone, Medin & Halberstadt, they sought to show non-monotonicity of similarity judgment, where adding a unique feature to one of the two objects being compared results in an increased similarity between the two. The intuition behind this is that the added feature brings to the foreground another dimension, along which the two objects are rather similar. (Consider the objects to be letters 'b' and 'd', for example — we are focusing on the shape of the letters here — and in the added feature part, we make 'd' with slightly thicker line.) Now, in fact, Goldstone and his colleague did not really find non-monotonicity, but they found that the similarity ratings were quite close in the two conditions. In our approach, we could perhaps say that bringing another dimension to the foreground has the effect of adding to the descriptor complexity, which counteracts the increased similarity along this new dimension. An interesting fact here is that in the follow up experiment, Goldstone *et al.* asked the subjects to explicitly write down the dimensions along which the two objects were similar, and they were able to find a small non-monotonicity effect. Perhaps explicitly writing down the dimension reduces some of the perceptual information load from the descriptor.

Conclusion and Future Research

In this paper, we outlined an algebraic formalization of SIT that allows domain-dependent operators to be used in structural decomposition of perceptual gestalts, and uses 'information load' to choose the preferred gestalt. We discussed how context effects on analogy and similarity can be modeled in this formalization, and then extended the framework to model categorization. In this extension, we characterized two complexity measures using the notion of information load: descriptor complexity, and member complexity. These two measures drive the categorization into opposite directions: from having a single category for all the objects to each object being in a category by itself. To find an optimum balance between these two extremes, we proposed a simple additive function; it remains to be seen how close this function comes to modeling the empirical data on categorization, and if other, more sophisticated, functions are necessary. Needless to say, much empirical research is necessary before we can start addressing such issues.

The use of the SIT algebra makes our approach perceptually motivated in that the optimum categories are those that would be perceived by a cognitive agent. Note that the notion of similarity on the basis of which the elements are categorized is defined in terms of similarity of structural descriptions of elements. The use of SIT operators together with the information load measure provide us a method in which similarity of elements is defined in terms of their perceptual similarity.

In this algebraic framework, it can be explained how new features can be created (Indurkhya 1997) and how they may affect the categorization. Different top-down goals of the cognitive agents bring to bear different sets of domain-dependent operations on the objects (and hence different algebras). In order to accomplish this, we may consider the SIT-algebra with a certain set of domain-dependent operations as the base algebra. Then, we may introduce the notion of sub-algebra or representation algebra by taking only a subset of domain dependent relations. Having a certain representation algebra which determines a certain categorization scheme, a new element will be interpreted according to the existing representation algebra and thus according to the existing categorization scheme. In this way, a new element may be interpreted differently than when it is viewed in isolation which means that new features of the element are detected. On the other hand, as the descriptor complexity and member complexity are both defined with respect to the representation algebra, a change in representation algebra causes changes in the metric in terms of which these two quantities are measured, which in turn affects which categorizations are considered optimum.

Finally, we conjecture that this model may also be applied to model semantics of non-perceptual concepts and categories. For instance, in an approach outlined in Gardenfors (1996), various objects are modeled as points in a multi-dimensional space, and categories (predicates) are made to correspond regions; it is argued that topological notions are more suited to capture the prototype effect observed in natural categories (Rosch 1978) — they can even provide a basis for analogies and metaphors across semantically distant domains. As topological notions can be treated algebraically, and the SIT notion of information load provides a metric to capture the idea of semantic distance between concepts, we believe that our approach is promising in this direction.

Acknowledgment

We wish to thank Remko Scha for the discussions and feedback during the development of the ideas presented in this paper.

References

Dastani, M. (1996) Computing the perceptual structure of visual expressions, *Proceedings of Artificial Intelligence in Design (AID'96), Workshop on Visual Representation, Reasoning and Interaction in Design.* 24-27 June, Stanford University, California.

Dastani, M. (1997) A Formal Framework for Visual Representations, em Proceedings of The 2nd International Workshop on Theory of Visual Languages, Capri, Italy, September '97.

Dastani, M. & Indurkhya, B. & Scha, R. (1997) An Algebraic Method for Solving Proportional Analogy Problems. *Proceedings of Mind II - Computational Models of Creative Cognition*, 15th - 16th September 1997, Dublin, Ireland.

Gardenfors, P. (1996) Conceptual spaces as a framework for cognitive semantics. In A. Clark et al. (eds.) *Philosophy and Cognitive Science*, Kluwer, Dordrecht, pp. 159-180.

Goldstone, R.L. (1994) Similarity, Interactive Activation, and Mapping. *Journal of Experimental Psychology: Learning, Memory, & Cognition 20*, pp. 3–28.

Goldstone, R.L., Medin, D.L., and Halberstadt, J. (1997) Similarity in Context. *Memory & Cognition 25* (2), pp. 237–255.

Hofstadter, D., and The Fluid Analogies Research Group (1995) *Fluid Concepts and Creative Analogies.* Basic Books, New York.

Indurkhya, B. (1991) On the Role of Interpretive Analogy in Learning. In *New Generation Computing, 8*, 385-402.

Indurkhya, B. (1997) On Creation of Features and Change of Representation. *Proceedings of the Japanese Cognitive Science Society's Special Interest Group on Language & Learning: Representation Change,* Vol. 97, No. 1, (SIGLAL 97-1), pp. 22–32.

Indurkhya, B. & Dastani, M. (1997) An Algebraic Model of the Context Effect in Analogy. *Proceedings of European Conference on Cognitive Science*, 239-242, April 9-11, 1997, Manchester, UK.

Leeuwenberg, E. (1971) A perceptual coding language for visual and auditory patterns. *American Journal of Psychology 84*, 307-349.

Pevtzow, R. and Goldstone, R.L. (1994) Categorization and the Parsing of Objects. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 717–722.

Rodet, L. and Schyns, P.G. (1994) Learning Features of Representation in Conceptual Context. *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, pp. 766–771.

Rosch, E. (1978) Principles of Categorization. In E. Rosch and B.B.Lloyd (eds.) *Cognition and Categorization*, Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 27-48.

Stins, J.F. and Van Leeuwen, C. (1993) Context Influence on the Perception of Figures as Conditional Upon Perceptual Organization Strategies. *Perception & Psychophysics 53,* (1), pp. 34–42.

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

Van der Helm, P. & Leeuwenberg, E. (1991) Accessibility: A criterion for Regularity and Hierarchy in Visual Pattern Code. *Journal of Mathematical Psychology, 35*, 151-213.