Estimating analogical similarity by vector dot-products of Holographic Reduced Representations.

Tony A. Plate
Department of Chemical Engineering
University of British Columbia
Vancouver, BC, Canada V6T 1Z4
tap@epi.terryfox.ubc.ca [1]

October 25, 1995

[1]This paper reports work performed in the Department of Computer Science at the University of Toronto

Abstract

Models of analog retrieval require a computationally cheap method of estimating similarity between a probe and the candidates in a large pool of memory items. The vector dot-product operation would be ideal for this purpose if it were possible to encode complex structures as vector representations in such a way that the superficial similarity of vector representations reflected underlying structural similarity. This paper describes how Holographic Reduced Representations (HRRs) [Plate 1991] provide such an encoding. HRRs are a method for encoding nested relational structures as fixed width distributed representations, i.e., vectors. The similarity of HRRs that can be computed quickly using dot-products. If two HRRs involve similar objects and relations, their dot-product will be high and will be sensitive to some aspects of structural similarity. However, if the two HRRs do not involve similar objects their dot-product will be lower and will not be as sensitive to structural similarities. This paper also describes an enrichment of HRRs called "contextualization". The dot-product of contextualized HRRs is sensitive to more aspects of structural similarity, and is also sensitive to structural similarity in the absence of shared object features.

1 Introduction

Gentner and Markman [1993] suggest that the ability to deal with analogy will be a "watershed or Waterloo" for connectionist models based on distributed representations. They identify "structural alignment", which is the matching of corresponding components of two structures in a manner which preserves the relationships, as the central aspect of analogy making, and note the apparent ease with which people can do this in a wide variety of tasks. The main impediment to building distributed connectionist models for tasks such analogy evaluation has been the lack of an adequate distributed representation for nested compositional structure. Gentner and Markman wonder whether the surface forms of distributed representations for structured objects would be similar to the degree that the underlying structures were similar, but express pessimism about the prospects for the development of such a representation.

In this paper I describe how Holographic Reduced Representations (HRRs) [Plate 1991; Plate 1995], which are a fixed-width distributed representation for nested compositional structures, provide such a representation. In particular, they can be used to obtain fast estimates of similarity between structures. A HRR is a high dimensional vector, and the vector dot-product of two HRRs is an efficiently computable estimate of the overall similarity between the two structures represented. This estimate is sensitive to surface similarity and some, but not all, aspects of structural similarity,[1] even though correspondences between structure elements are not explicitly computed. Comparisons of HRRs can be made more sensitive to structural similarity by including additional bindings in the HRRs, which are derived using a technique I call "contextualization".

Gentner and Markman [1993] and Gentner, Markman and Wisniewski [1993] remark that people appear to perform structural alignment in a wide variety of tasks, including perception, problem solving, and memory recall. One task they and others have investigated is analog recall. A subject reads a number of stories and later reads a probe story. The task is to recall stories that are similar to the probe story (and sometimes evaluate the degree of similarity and perform analogical reasoning). This task is often thought to involve two stages – first is recall from long-term memory of stories that are potentially relevant and analogical to the probe story, and second is evaluation and use of the analogy, which involves construction of mappings between the recalled stories and the probe story. Two computer models of this process, Thagard *et al*'s [1990] ARCS (Analog Retrieval by Constraint Satisfaction) and Gentner and Forbus's [1991] MAC/FAC (Many Are Called, Few Are Chosen), also have two stages. The first stage selects a few likely analogs from a large number of potential analogs. The second stage searches for an optimal (or at least good) mapping between each selected story and the probe story and outputs those with the best mappings. MAC/FAC uses Falkenhainer, Forbus and Gentner's [1989] SME (Structure Mapping Engine) as its second stage. The two-stage architecture is illustrated in Figure 1. The first stage selects potential analogies to the probe from the large pool of items in long-term memory, using fast but superficial comparisons. The bottleneck depicted here is typical but not necessary – the computation required by the first stage could happen in parallel throughout long-term memory (e.g., in a content-addressable memory). The second stage performs a

[1] "Surface features" of stories are the features of the entities and relations involved, and "structural features" are the relationships among the relations and entities.

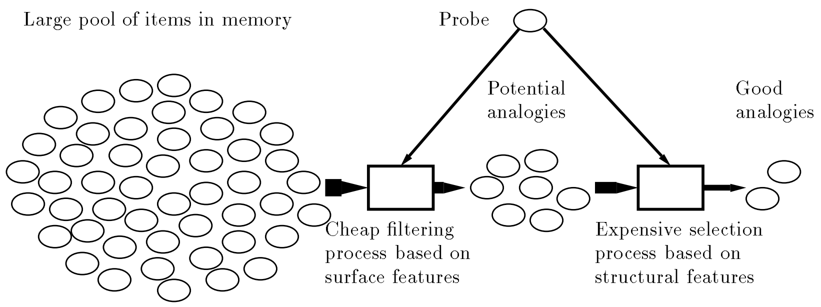

Figure 1: Two-stage models of analogy retrieval.

more expensive structural comparison between the probe and each potential analogy passed by the first stage. Two stages are necessary because it is too computationally expensive to search for an optimal mapping between the probe and all stories in memory. An important requirement for a first stage is that its retrieval speed scale well with both the size and number of episodes in long-term memory. This requirement prevented the authors of MAC/FAC and ARCS from having the first stage consider structural features, because they knew of no efficient way to do this.

The evidence that people take structural correspondences into account when evaluating and using analogies is longstanding and solid (see Gentner, Rattermann and Forbus [1993] for a review). However, it is less certain whether structural similarity influences access to long-term memory (i.e., the first-stage reminding process). Gentner and Forbus [1991] and Gentner, Rattermann and Forbus [1993], found little or no effect of analogical similarity on reminding. Others, e.g., Wharton *et al* [to appear], Seifert and Hammond [1989] and Ross [1989], have found some effect. Wharton *et al* [to appear] suggest that the experimental design in earlier studies made it difficult to separate the effects of surface and structural similarity. In any case, surface features appear to influence the likelihood of a reminding far more than do structural features. Ross's [1989] study suggested that the effect of structural similarity on reminding is greater in the presence of surface similarity.

The first goal of this paper is to introduce HRRs as a distributed representations for nested compositional structure.

The second goal is to investigate how the dot-product of HRRs performs as an estimate of analogical similarity and to spell out the strengths and weaknesses of this estimate. To this end I show how dot-products of HRRs respond to various surface and structural aspects of similarity and compare the similarity judgements to those of ARCS and SME, two widely cited models of analogy evaluation and analogical reasoning. Thagard *et al*'s [1990] ARCS builds a local connectionist constraint satisfaction network (in its second stage) whose stable states are good interpretations of analogy between two structures. An interpretation of an analogy is a mapping of the entities in one structure to those in the other. The network is allowed to settle to a stable state and an overall similarity score is derived from how well the various constraints are satisfied. Falkenheimer, Forbus and Gentner's [1989] SME incorporates an explicit search for optimal analogical mappings. Both ARCS and SME

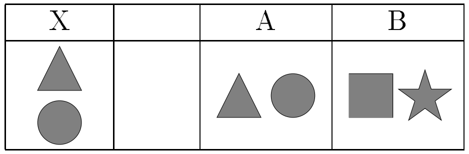

Figure 2: An example of the shape-configuration task.

provide an explicit interpretation of an analogy as well as a measure of goodness, and both can use the interpretation to perform analogical reasoning.

The third goal is to suggest that dot-products of HRRs could be an efficient mechanism for the first stage of a model of analog retrieval. This would be more a powerful mechanism than the current first stages of both ARCS and MAC/FAC because it would be sensitive to structural as well as surface similarity. However, a second stage would still be required for accurate evaluations of structural similarity, because the HRR dot-product is not an infallible measure of structural similarity. Furthermore, analogical reasoning tasks require an interpretation of an analogy, and this is not provided by HRR dot-products.

I use two sets of examples to illustrate how analogical similarity can be estimated by dot-products of HRRs. The first is a set of shape configurations from the paper by Markman *et al* [1993]. Figure 2 shows an example; the task is to judge which of A or B is more similar to X. These examples can each be represented by a single relation such as `above(triangle,circle)`. I include this set because it is a good illustration of some of the surface and simple structural features that people take into account when judging similarity. The similarity rankings given by HRR dot-products agree with those given by people.

The second set of examples consists of one-sentence episodes such as the following:

> "Spot bit Jane, causing Jane to flee from Spot."
> "Fido bit John, causing John to flee from Fido."
> "Rover bit Fred, causing Rover to flee from Fred."

The episodes in this set can be represented by two or three nested relations, e.g.:

`cause(bite(spot,jane),flee(jane,spot))`.

I use these examples to illustrate the extent to which HRRs are sensitive to the more complex aspects of structural similarity, such as patterns of variable instantiation and structural arrangement of propositions. It turns out that ordinary HRRs are only sensitive to structure in the presence of similarity among corresponding objects. However, HRRs can be made sensitive to structural similarity by incorporating contextual information.

In Section 2 of this paper, I describe HRRs, which are a distributed representation for nested compositional structure. In Section 3, I describe how HRRs can be applied to the

shape-configuration examples. In Section 4, I describe how HRRs can be used to judge the similarity of episodes involving multiple, nested propositions. In Section 5, I describe how interpretations of an analogy, i.e., corresponding objects, can be extracted from HRRs. In Section 6, I discuss various issues, such as the relationship of HRR similarity ratings to ratings given by SME and ARCS, scaling to larger and more examples, flexibility of comparison, and pros and cons with respect to other distributed representations for nested compositional structure.

2 Holographic Reduced Representations (HRRs)

The difficulty of representing recursively nested structures has been an obstacle to the use of distributed representations and neural networks for higher-level cognitive tasks such a reasoning and language processing [Fodor and Pylyshyn 1988; Smolensky 1990; Hinton 1990]. There have been a number of methods proposed for forming associations in distributed representations in a manner able to handle recursively nested structure. These include Tensor Products [Smolensky 1990], RAAMs [Pollack 1990], and BoltzCONS [Touretzky 1990]. Holographic Reduced Representations (HRRs) are a distributed representation for nested compositional structures, which retain nearly all the advantages of distributed representations for objects. HRRs have much in common with Smolensky's tensor-products representation for propositions, but use circular convolution to associate roles and fillers, rather than the tensor product. The benefit of this is that the dimensionality of representations remains constant rather than increasing exponentially with the depth of composition as with tensor products. Convolution has been used as the associative operator in a number of computational and psychological memory models [Willshaw, Buneman and Longuet-Higgins 1969; Borsellino and Poggio 1973; Liepa 1977; Willshaw 1981; Murdock 1982; Metcalfe Eich 1982; Murdock 1987; Paek and Psaltis 1987].

In this Section, I give an overview of HRRs; for more a detailed discussion, see Plate [1995] or Plate [1994].

2.1 Advantages of distributed representations

Distributed representations [Hinton, McClelland and Rumelhart 1986] are commonly described as patterns of activity over a large number of computing elements (neurons). For the purposes of this paper, I use an abstract version of this description: a distributed representation is a vector of real values. The number of elements in the vector (i.e., the vector dimension) is usually fixed.

Distributed representations are attractive for a number of reasons:

- There is a simple and computationally efficient measure of similarity between two distributed representations – the vector dot-product.[2] In general, similar items can

[2] The vector dot-product of two n-dimensional vectors $\mathbf{x}$ and $\mathbf{y}$ is defined as: $\mathbf{x}\cdot\mathbf{y} = \sum_{i=0}^{n-1} x_i y_i$ where x_i and y_i are the i-th elements of $\mathbf{x}$ and $\mathbf{y}$. I assume that vectors are normalized, i.e., their Euclidean length is 1. If not, the cosine is often a better measure of similarity.

be represented by vectors which have a high dot-product.

- Items are represented in a continuous space. This has two benefits:
  - It allows similarities to vary continuously.
  - Representations can be learned using gradient descent techniques.
- The information in the vector is distributed and redundant. This has several useful consequences:
  - Multiple items can be stored in a single buffer, by superimposing their representations. The number of items that can be superimposed is a soft-limit – items become degraded as more items are added.
  - The representations degrade gracefully as noise is added.
  - Representations which are degraded either by noise or by superposition can be cleaned up using an error-correcting associative memory.
- They can be processed in a parallel network of simple processing elements

2.2 Representational assumptions

The basic representational assumptions used in this work (and related work on distributed connectionist representations for nested structure) are as follows:

- All entities (including those with complex structure) are represented by fixed-width vectors.
- The representation of a compositional entity is derived, in a systematic fashion, from the representations of its components.
- The representation of an atomic entity (i.e., one not composed of other entities) is arbitrary. In HRRs, atomic or base-level entities are represented by random vectors.

2.3 Associations and complex structure

For more complex tasks we require a way of representing associations between items. Suppose we have distributed representations for the concept "circle", "triangle", "small", and "large". We can represent a small circle by superposing the patterns for "small" and "circle". However, when we try to represent a small circle and large triangle we have the problem that the superposition of the four patterns is ambiguous – the information that small is associated with triangle and large associated with circle is lost – it could be a large circle and a small triangle. The same problem arises when we try to represent conceptual relations. Suppose we have a predicate representation for "Spot bit Jane": `bite(spot,jane)`. Spot is the agent of this relation, and Jane is the object (or patient). A distributed representation of this relation must be careful to preserve the information about which person is associated with which role (agent or object) so that there is no confusion with "Jane bit Spot". I refer to associations between roles and fillers as *role/filler bindings*.

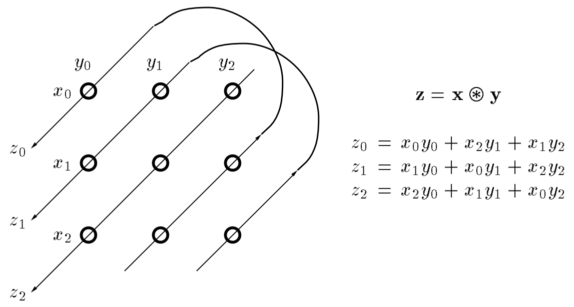

Figure 3: Circular convolution represented as a compressed outer product.

One obvious way to avoid ambiguity about who is associated with which role is to divide the vector into several blocks, and devote a block to each role [Hinton 1990]. The drawback of this method is that it is unsuitable for representing recursively nested relations such as "Spot bit Jane, causing Jane to flee from Spot": `cause(bite(spot,jane),flee(jane,spot))`. An alternative is to use a function to combine two vectors (representing a role and a filler) into one. This is the approach taken by Smolensky [1990] with tensor products and Pollack [1990] with RAAMs and is also the approach taken with HRRs, though in each case the function used to combine vectors is different.

If the method for associating roles and fillers can be applied recursively, as it can with tensor products, RAAMs and HRRs, then it can represent nested structures. The main issue of concern in this paper, and the issue which concerned Gentner and Markman [1993] is whether the structure of complex objects is reflected in the surface form of the resulting representation.

2.4 Circular convolution

Circular convolution is an operation that maps two vectors onto one, i.e., $\circledast : \mathcal{R}^n \times \mathcal{R}^n \to \mathcal{R}^n$. If $\mathbf{x}$ and $\mathbf{y}$ are n-dimensional vectors (subscripted 0 to $n-1$), then the elements of $\mathbf{z} = \mathbf{x} \circledast \mathbf{y}$ are

$$z_i = \sum_{k=0}^{n-1} x_k y_{i-k}$$

where subscripts are taken modulo-n. Circular convolution can be viewed as a compression of the outer (or tensor) product of the two vectors, as Figure 3 shows. Each of the small circles represents an element of the outer product of x and y, e.g., the middle bottom one is x_2y_1. The elements of the circular convolution of x and y are the sums of the outer product elements along the wrapped diagonal lines.

Circular convolution can be regarded as a multiplication operator for vectors. It has many properties in common with both scalar and matrix multiplication. It is commutative ($\mathbf{x} \circledast \mathbf{y} = \mathbf{y} \circledast \mathbf{x}$), associative ($\mathbf{x} \circledast (\mathbf{y} \circledast \mathbf{z}) = (\mathbf{x} \circledast \mathbf{y}) \circledast \mathbf{z}$), and bilinear ($\mathbf{x} \circledast (\alpha\mathbf{y} + \beta\mathbf{z}) = \alpha\mathbf{x} \circledast \mathbf{y} + \beta\mathbf{x} \circledast \mathbf{z}$). There is an identity vector $\bar{I}$ ($\bar{I} \circledast \mathbf{x} = \mathbf{x}$) and a zero vector $\bar{0}$ ($\bar{0} \circledast \mathbf{x} = \bar{0}$). Inverses $\mathbf{x}^{-1}$ exist