We want computer systems that can help us assess the similarity or relevance of existing objects (e.g., documents, functions, commands, etc.) to a statement of our current needs (e.g., the query). Towards this end, a variety of similarity measures have been proposed. However, the relationship between a measure's formula and its performance is not always obvious. A geometric analysis is advanced and its utility demonstrated through its application to six conventional information retrieval similarity measures and a seventh spreading activation measure. All seven similarity measures work with a representational scheme wherein a query and the database objects are represented as vectors of term weights. A geometric analysis characterizes each similarity measure by the nature of its iso-similarity contours in an n-space containing query and object vectors. This analysis reveals important differences among the similarity measures and suggests conditions in which these differences will affect retrieval performance. The cosine coefficient, for example, is shown to be insensitive to between-document differences in the magnitude of term weights while the inner product measure is sometimes overly affected by such differences. The context-sensitive spreading activation measure may overcome both of these limitations and deserves further study. The geometric analysis is intended to complement, and perhaps to guide, the empirical analysis of similarity measures.

1. Introduction

The success of any information retrieval system depends upon its ability to accurately assess the relevance of objects (e.g., information units, documents, functions, commands, etc.) in its database to a given user's request. Towards this end, components of a ranking algorithm include a means of representing the user's request, a means of representing the objects of the database, and a means of measuring the similarity between these representations [1]. This article focuses on the last of these components, the ranking algorithm's similarity measure.¹

A wide range of similarity measures has been proposed (e.g., [1–4]) and the number of potential similarity measures is larger still [1]. Yet, surprisingly little is known about the comparative utility of these measures. Through a combined use of formal, algebraic manipulation and empirical simulation, McGill, et al. [3] were able to reduce a set of 67 similarity measures to a set of 24 classes. The members of a class were judged either to be algebraically equivalent to one another or else to exhibit very high correlation in their judgments of similarity. However, the results of an experimental comparison of measures from various classes were inconclusive.

McGill, et al. [3] (pp. 10–19), acknowledge a number of difficult methodological problems that severely limit our ability to generalize from empirical studies of similarity measure performance. A number of problems arise, for example, in the attempt to judge the precision and recall rate of a similarity measure (see [4] pp. 157–191). Moreover, no means has yet been proposed to systematically and economically manipulate various important factors of the information retrieval task including: the nature of the database and its size, the nature of the user population and its information needs, the procedures for arriving at representations of database objects and user requests. It is quite possible that these factors interact with the choice of similarity measure so that a performance ordering of similarity measures may change depending upon the choices that are necessarily made with respect to each of these factors. It is also possible that important differences among similarity measures are "averaged away" in studies that allow these factors to vary in an uncontrolled fashion.

¹The phrase "relevance measure" is perhaps a more accurate label for this component to a ranking algorithm since its primary purpose is to assess an object's relevance to a user's request. The computer's representation of an object could conceivably bear very little resemblance to its representation of a user's request and yet still be very relevant. "Similarity measure" is used instead throughout this article to maintain consistency with its usage in the information retrieval literature.
As a complement to the empirical analysis of similarity measures and their effectiveness, this article advances a geometric method of analysis. We illustrate the use of this approach through its application to six conventional vector similarity measures and we also look at a seventh spreading activation similarity measure derived from proposed mechanisms of retrieval in human memory [5–9]. The geometric analysis gives us visualizations that reveal major differences among the similarity measures under consideration. We emphasize here at the outset that the practical significance of any such differences must be empirically validated; the geometric approach cannot take the place of such empirical validation. However, the geometric approach may help to complement, and perhaps to guide, empirical analyses of similarity measures. Specifically, the geometric approach described in this article may help us to better understand the circumstances in which algebraic differences in similarity measure formulas will have an empirically verifiable impact upon retrieval performance.

We begin by setting up the fundamentals: the representation of terms, objects, and queries, in both an algebraic and a geometric (graphical) framework. We then introduce the use of iso-similarity contours as a means of geometrically characterizing the actions of a given similarity measure. With these fundamentals established, we will work through the seven similarity measures, elaborating the basic concepts as needed.

2. Fundamentals

The Basic Algebraic Representation

The similarity measures considered in this article require that both the user's query and the objects in the database be represented as collections of weighted terms. In standard library information retrieval, objects might be journal articles. The terms of an article and their weights might have been manually assigned by a human indexer based upon a topic classification scheme or these might have been automatically derived from frequency counts of content words in the article itself. Alternatively, under an adaptive indexing scheme [10], terms and term weights might have been dynamically compiled from the observed attempts of people to access objects (e.g., on-line entries from a classified "yellow-pages" section or from a command documentation manual).

When considering the collection of objects as a whole, it is convenient to place the weights in a matrix, as in Table 1. The terms from the full collection are associated with the rows of the matrix. The objects are associated with the columns. The weights appear in the cells of the matrix, with entry \( w_{ij} \) indicating the weight with which term \( T_i \) is associated to object \( O_j \). (\( w_{ij} \) is zero when the term is not relevant.) Thus the weighted term list for an object corresponds to a column vector of this matrix.

A concrete situation and example will be useful. Suppose that you have a small orchard and are seeking information about care for your fruit trees. Suppose, further, that the Department of Agriculture has a set of 6 publications on apple trees and orange trees, represented in Table 2. Publication \( O_1 \), with an Apple weight of 1 and an Orange weight of 0, might be an introductory pamphlet on apple trees only. Publication \( O_6 \), with an Apple weight of 8, might be a much more complete and exhaustive treatise on apple trees. Publication \( O_4 \) would be the corresponding exhaustive work exclusively about orange trees. Publication \( O_4 \) might be a moderately complete work on both apple and orange trees, equally. And so on.

Implicit in the preceding example are at least two important aspects of a term weight:

- The term weight \( w_{ij} \)'s magnitude relative to other term weights in column \( j \) can express the importance of term \( i \) in the representation of object \( O_j \) relative to other terms in the object's representation. This within-object comparison is important because it tells us the object's semantic content, or topic area. Thus for publication \( O_1 \), comparing Apples = 1 with Oranges = 0 leads to the conclusion that the publication is exclusively about apple trees.
- The term weight \( w_{ij} \)'s magnitude relative to other term weights in row \( i \) can express object \( O_i \)'s relevance to term \( T_i \) (i.e., the intensity of its \( T_i \) content) relative to other objects containing term \( T_i \) in their representation. Thus, looking between objects just within the Apple row, we can see that Publication \( O_2 \), whose Apple weight is only 1, Note that this is true even though \( O_2 \) is an "all-apple" publication and \( O_4 \) is an "apple + orange" publication.

### Table 1. A matrix representation of the objects in a database.

<table>
<thead>
<tr>
<th>Terms</th>
<th>( O_1 )</th>
<th>( O_2 )</th>
<th>( O_3 )</th>
<th>( O_4 )</th>
<th>( O_5 )</th>
<th>( O_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T_1 )</td>
<td>( w_{11} )</td>
<td>( w_{12} )</td>
<td>( w_{13} )</td>
<td>( w_{14} )</td>
<td>( w_{15} )</td>
<td>( w_{16} )</td>
</tr>
<tr>
<td>( T_2 )</td>
<td>( w_{21} )</td>
<td>( w_{22} )</td>
<td>( w_{23} )</td>
<td>( w_{24} )</td>
<td>( w_{25} )</td>
<td>( w_{26} )</td>
</tr>
<tr>
<td>( T_3 )</td>
<td>( w_{31} )</td>
<td>( w_{32} )</td>
<td>( w_{33} )</td>
<td>( w_{34} )</td>
<td>( w_{35} )</td>
<td>( w_{36} )</td>
</tr>
<tr>
<td>( T_4 )</td>
<td>( w_{41} )</td>
<td>( w_{42} )</td>
<td>( w_{43} )</td>
<td>( w_{44} )</td>
<td>( w_{45} )</td>
<td>( w_{46} )</td>
</tr>
<tr>
<td>( \cdots )</td>
<td>( \cdots )</td>
<td>( \cdots )</td>
<td>( \cdots )</td>
<td>( \cdots )</td>
<td>( \cdots )</td>
<td>( \cdots )</td>
</tr>
</tbody>
</table>

### Table 2. A matrix representation of an "apples and oranges" database.

<table>
<thead>
<tr>
<th>( O_1 )</th>
<th>( O_2 )</th>
<th>( O_3 )</th>
<th>( O_4 )</th>
<th>( O_5 )</th>
<th>( O_6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apples</td>
<td>2</td>
<td>1</td>
<td>6</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Oranges</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>
In this manner, a term weight can be said to have both a within-object and a between-objects function.

Of course it is possible to glean other information from database matrices of term weights such as those depicted in Tables 1 and 2. The discrimination value of a term \( T_i \), for example, might be inferred from its pattern of usage in the database, where this is represented by term \( T_i \)’s associated row vector in the database matrix [4] (pp. 59–71) [11–14]. It is also possible sometimes to draw inferences concerning the similarity or synonymy of terms in the database by comparing their patterns of database usage. This supposition underlies various techniques of automatic thesaurus construction [4] (pp. 75–81) [15–17]. And techniques to elicit the latent structure of a database matrix promise to realize substantial improvements in the performance tradeoff between recall rate and precision [18]. We will discuss similarity measure sensitivity to these other kinds of database information later in this article.

A few final notes about the matrix of term weights. The example matrix we have shown has various numerical values for entries. We simply note that in the boolean case, these would be limited to 0 and 1, representing simply the absence vs. presence (or irrelevance vs. relevance) of a given term for an object. While this is not uncommon in the literature, we will generally be considering the more general, non-boolean case. We will, however, assume that all term weight components of a given vector are greater than or equal to zero; object vectors are thus component-wise non-negative (or simply non-negative).

The Basic Geometric Representation

Corresponding to the matrix representation for objects and terms there is a geometric representation of central interest to this paper. Geometrically, the objects in the weight matrix can be represented by points in a space, using an object’s term weights (in its column vector of the matrix) as coordinates. We can also graphically represent an object’s vector by simply drawing an arrow from the origin to the object’s point in the space. The six Apple and Orange documents of Table 2 are shown geometrically in Figure 1.

The two “apple-only” publications \( O_1 \) and \( O_2 \) thus lie on the horizontal (Apple-weight) axis, at \((1,0)\) and \((8,0)\) in the space; the “apple + orange” document \( O_4 \) lies at \((3,3)\) in the space; etc.

The geometric depiction of an object vector makes salient the features of direction and length, each of which has a corresponding semantic interpretation with reference to information that is potentially present in a database of term weights. The direction of an object’s vector is determined by the within object relationships of its associated term weights and is, for that reason, an indication of semantic content or topic. In turn, the length of an object vector, relative to other vectors, is determined by between-object relationships of its associated term weights and is, for that reason, an indication of the object’s topic intensity.\(^2\) For expositional purposes we

\(^2\)Of course, there are different notions of intensity corresponding to different interpretations of the between-object differences in term weights.

FIG. 1. A geometric representation of the “apples and oranges” database.

will primarily work in a 2-space of object vectors, as in Figure 1. However, we will discuss the ways in which conclusions drawn from this special two-dimensional case generalize to the high number of dimensions required for a realistic number of terms. Since term weights are constrained to assume values greater than or equal to zero, we will focus primarily on the first quadrant of this 2-space. Again, however, many observations we make with respect to this subspace readily generalize to situations in which object and query term weights can assume negative real values as well.

The Representation of Queries

Under the framework provided here, a query is presumed to be very much like an object. A given query has a matrix algebraic representation as a vector of term weights and it has a corresponding geometric representation as a point in a space of possible term weight combinations. As in the case of objects, we can draw an arrow from the origin to the query point in order to highlight properties of length and direction. The query might be viewed as a description or profile of the ideal object. Alternatively, it can be viewed as a specification of the relative importance of terms for a retrieval attempt. Following the Apples and Oranges example, if you had only an apple orchard then your query might have some significant Apple weight, but an Orange weight of 0. By contrast, if you were growing both apples and

Intensity might correspond to the sheer quantity of an object’s content or it might represent some assessment of the quality of this content. Intensity might also correspond to the “accessibility” of an object’s content such that an information object with a high intensity rating is intended primarily for the experts of a field. If we consider, for example, the relative vector lengths of an unedited doctoral thesis, a carefully crafted technical journal article, and an introductory newspaper article, all over the same topic, these are likely to vary depending upon which interpretation of intensity is adopted during the assignment of term weights. As we will shortly see, similarity measures also vary in their interpretation of object intensity.
oranges, your query might be equally weighted along the Apple and the Orange dimensions. The advantage of this representation of queries is that it is identical to that for objects, so that a direct comparison is possible. Retrieval proceeds by comparing a query to the objects in the database, and selecting those objects whose representations are in some sense similar to the query's.3

In our discussion of the similarity measures we will make frequent reference to the diagonal which is the line coincident with object O4 in Figure 1. In n-space this is the line coincident with a vector all of whose components are equal. One significance of the diagonal, for our purposes, is that points along it can be used to represent queries in which interest is evenly divided among objects. (These points can also be used, of course, to represent objects whose topical content is evenly distributed among the columns of the database.)

The Geometric Representation of Similarity Measures

A similarity measure yields, for any two vectors, a numerical value indicating how similar the two vectors are. We will investigate a given similarity measure by taking a fixed query vector and mapping out the similarity value assigned to various regions of the space. One of the most useful ways to draw such a map is in analog to a topographic map in geography — using contour lines. Such lines connect points that have equal value (equal altitude in topography, equal similarity to the query in our investigations). The "iso-similarity" contours show what the similarity measure is sensitive to: moving an object from one contour to the next will greatly effect its similarity to the query; moving it along the contour will make no difference. This will become clearer in the first example of the next section, but suffice it to say that the different similarity measures will have quite different pictures — indicating different sensitivities. Given the meaning of direction (as topic) and length (as topic intensity) in the space, the different sensitivities reflect the various merits of the measures.

3 Another interesting consequence of the comparability of object and query representations is that the column vector of any object Oi could itself be used as a query vector. A reasonable interpretation might then be that there is interest in assessing the similarity of other objects in the database to object Oi.

3 iso-similarity contours in this article's figures were initially drawn using the ISOPLOT program, written by George Furnas in Symbolics Zeta-LISP, 1986. Copies may be obtained by writing Dr. George W. Furnas, Bell Communications Research, 435 South St., Morristown, NJ 07960.

The Inner Product Measure

One of the most basic similarity measures is the dot or inner product measure. Its importance stems from its algebraic simplicity and from its use as a component in many other similarity measures (including five of the six remaining similarity measures to be considered in this article). The formula for the inner product measure of similarity between a query Q and object O is:

$$\sum_{i=1}^{n} w_{Qi} \times w_{Oi}$$

This formula is familiar to any student of linear algebra, and linearity is its key feature. The formula is a simple weighted sum. As a consequence of its algebraic linearity an increment in the weight of any term in an object’s representation will have a proportional effect on the object’s similarity rating. Moreover, the contributions of different terms are quite independent of each other. In a boolean case, where term weights in both the query and object representations can assume only the binary values of 0 and 1, the inner product can be seen to compute the cardinality of the intersection between the set of terms associated with the query and the set of terms associated with the object. In a general situation, where query and object term weights can assume any value, the inner product can be seen to attempt a prediction of an object’s relevance to a user’s request (the dependent measure) on the basis of a weighted, linear combination of object term weights (the independent measures). As we will shortly see, the inner product is often at the heart of other similarity measures where it works as a basic comparison operation that is then normalized in various ways by other components in the measure.

The inner product’s feature of linearity is readily apparent in its geometric representation. Figure 2 shows some sample iso-similarity contours of the inner product. These contours are sets of evenly spaced, straight lines, perpendicular to the arrow representing the query vector. All points falling on the same contour line have the same inner product similarity to the query. Thus for example, points A, B, and C (see Figure 2a) all have the same inner product measure and this is smaller than that shared by the points D, E, and F.

If there were n terms instead of two, we would have to use an n-dimensional space, and the inner product iso-similarity contours would be n-1 dimensional hyperplanes rather than the lines of Figure 2. However, everything would still be linear: hyperplanes would be evenly spaced.
and perpendicular to the query vector. In general, the inner product measure can be seen to induce an orthogonal projection of object vectors onto a new dimension defined by the query vector; the larger an object's value on this dimension, the higher will be its similarity weighting. This means that all object vectors lying on a given $n{-}1$ dimensional hyperplane (e.g., a line in 2-space) that is perpendicular to the query vector will receive the same similarity rating.

The iso-similarity contours, like those of Figure 2 have many uses.

- Given two objects in the space, it is possible to simply look and decide which is higher in similarity to the query, and by how much. This is exactly analogous to using a topographic map to decide which of two places is higher in altitude.
- Closely related to this, the contours tell how one could change a given object to achieve some desired effect on its similarity: move along the contour for no change, move perpendicular to the contour towards the next higher contour to maximally increase its similarity, or move perpendicular to the contour towards the next lower contour to maximally decrease its similarity.

FIG. 2. Sample iso-similarity contours for the inner product.
is exactly analogous to using a contour map to find which way is uphill and downhill from where you stand.

- One can choose an arbitrary place in the space and see what similarity would be obtained by a new object introduced at that place.
- The contours can be used to illustrate points of maximal and minimal similarity, should they exist. In the inner product case, such extrema do not exist (except as introduced by our boundary, at the origin.) Finding such extrema is analogous to finding peaks and valleys in geographic contours.
- As mentioned in the introductory section, the geometric notions of direction and length (or radius) have an important interpretation in information retrieval as topic and intensity (as these are determined by within-object and between-object term weight relationships). The contours help to illustrate how these geometric notions are captured by a given measure.

What does all this mean in the world of objects and queries? A geometric characterization of the inner product in terms of these iso-similarity contours helps to illustrate several important attributes with direct implications for its use as a similarity measure:

- **Angle monotonicity**—Given any two object vectors of equal L2 or Euclidean length, the inner product gives the higher similarity rating to the object vector with a smaller angle of separation from the query vector. For example in Figure 3a, vectors A, B and C fall on successively higher contours, as their angular separation from the query Q decreases. Geometrically, this means that, taking any circle centered at the origin, as you move away from the query on that circle, the contours go down. Remembering that the interpretation of direction is something like “semantic content” or “topic”, this means that the inner product has the property that objects are preferred if they are closer to the query’s own topic, all else being equal (i.e., that aspect represented by the length of the vector).

- **Radial monotonicity**—Given any two object vectors with the same direction and, consequently, the same angle of separation from a given query vector, the inner product gives the longer vector a similarity rating larger than or equal to that of the shorter object. Geometrically, this means that if you start at any point in the space and move away from the origin along a line parallel to any axis, you will eventually cross arbitrarily large contour lines. This means that an object with very little overall to do with the query’s topic (and, consequently, with a vector representation at a comparatively wide angle of separation from the query vector) may, nevertheless, be rated as more similar to the query than all other objects in the database. In Figure 3d, for example, vector B will receive a larger similarity rating than vector A even though vector A has a comparatively much smaller angle of separation from the query.

- **Boundedness of similarity rating**—A final property of the inner product measure we will discuss is simply the range of its possible values. When query and object vectors are constrained to be non-negative, the inner product similarity ratings are bounded from below by zero. However, there is no upward bound on the magnitude of a similarity rating. That is, if the picture were extended far enough, the values on the contours would get arbitrarily large. One related property of an un-

<table>
<thead>
<tr>
<th>Table 3. If a similarity measure is subject to unbounded single-component influence, it is possible to achieve an arbitrarily high similarity rating through manipulations in w.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>6</td>
</tr>
</tbody>
</table>

---

5Strictly speaking, this is true only when the angle of separation between the query and the object vector is less than 90 degrees. If the two are perpendicular, so that the angle is exactly 90 degrees, length has no effect, and if the angle is more than 90 degrees the effect of length is monotone decreasing. This last case cannot occur, of course, if we restrict all weights to be non-negative.

6Again, strictly speaking, this is not true when the direction of change is at right angles to the query vector. This will occur when the direction of change corresponds to a zero-valued query term.

bounded similarity range is that there is no notion of an “ideal object,” no notion of a “perfect score”; i.e., there is no peak or maximum in the contour set.

The inner product’s attribute of angle monotonicity indicates some sensitivity to topic as expressed through within-object (and within-query) term weight relationships. Among object vectors of equal length, the measure will rate more highly those vectors that are more nearly in the direction of the query’s vector representation. On the other hand, the inner product’s attribute of component-wise (and hence radial) monotonicity reflects a sensitivity to intensity as expressed through between-object term weight relationships. The inner product measure favors objects that are more heavily weighted along term dimensions of the query. Moreover, it does not penalize an object for its “representational richness”, i.e., for increases in the magnitude of non-query (i.e., zero-valued) term weights in the object’s representation.

The usefulness of the inner product’s attributes of component-wise and radial monotonicity depends upon the appropriateness of the “More is better” aphorism. The most
questionable attribute of the inner product is probably the unboundedness of single component influence. As a result of this attribute, objects centered quite far off the query topic can still be rated as highly relevant to a query. In the next section, we will consider the cosine measure which differs on precisely these debatable points.

**The Cosine Measure**

The formula for the cosine measure of similarity⁷ is:

$$\frac{\sum_{j=1}^{n} W_{Qj} \cdot W_{Oj}}{\sqrt{\sum_{k=1}^{n} (W_{Qk})^2} \cdot \sqrt{\sum_{j=1}^{n} (W_{Oj})^2}}$$

The numerator of the cosine measure is itself nothing more than the inner product measure discussed in the previous section. The critical difference, then, is that the inner product is divided by the product of the L2 or Euclidean lengths of the object and query vectors. This division may be best understood as a normalization.

Notice that the formula may be re-written as the inner product of the vectors $Q^*$ and $O^*$, where these vectors are derived by normalizing the query and object vectors, respectively, so that their L2 lengths each equal one. Thus the measure can be conceived of as first normalizing, then comparing the objects. The net result of the cosine measure may be understood by analyzing the effects of its component operations and how these effects combine.

Let us first consider the normalization. This L2 length normalization, like other normalizations, has the effect of ignoring some specific kind of difference.⁴ In this case, any vectors $O_1$ and $O_2$ with the same direction and differing only in length, will be transformed by this normalization into the same vector $O$, of L2 length 1. The result in 2-space is that the normalized object vectors all lie upon a circle of radius 1 from the origin. In higher dimensions the points all lie on the surface of a hypersphere of radius 1.⁸ The subsequent comparison operation will only work on these normalized points. The critical fact is that, in any subsequent comparison, all points anywhere in the space are treated exclusively in terms of their special normalized representative on this unit sphere. Since all points on a given line radiating from the origin have the same normalized representative, all points on such a given line will have to receive the same similarity value from the comparison. Thus we have our first insight into what the iso-similarity contours will look like: Regardless of the subsequent comparison process, at least the points on the same radial line will be iso-similar. Figure 4a shows this aspect of the iso-similarity contours resulting from the normalization. Note that we cannot yet assign values to these contours, since we have not made any comparisons. We simply know that since all points on one of these lines normalize to the same unit vector, they will all have the same value, whatever that turns out to be. Note also that we do not know if some of these lines will have the same values as each other—that is, additional iso-similarity may arise from the comparison operation.

Let us temporarily forget the lines of normalization, and only concentrate on the similarity of the various normalized unit vectors to the query. In the case of the cosine measure, the normalized vectors are compared to the query via the inner product. From the previous section we know that two points will have the same inner product similarity with respect to the query if and only if they are on the same inner product iso-similarity line (or hyperplane, in the higher dimensional case). Thus two of our normalized vectors lying on their unit circle (or unit hypersphere) will be iso-similar if and only if they lie at places where a given inner product line (hyperplane) intersects the circle (hypersphere). Figure 4b shows these intersections in 2-space: pairs of points are on either side of the query vector. The net result is that two unnormalized vectors are equally similar to the query if they are on either of the normalization lines through these pairs of points. That is, the iso-similarity contours are pairs of radial lines centered around the query (with a spacing that decreases as the angle from the query increases). As Figure 4c illustrates, regions in which vectors have roughly the same similarity rating (to within 0.025 in the figure) are actually wedge-shaped, indicating that an object vector’s similarity rating depends entirely on its angle of separation from the query vector. The wedge-shaped, radial character of this picture comes from the normalization; the pairing (of regions of comparable similarity on either side of the query vector) and spacing come from the inner product comparison.

In three-space, the normalization lines still radiate from the origin, but the inner product comparison picks out points on the intersection of the inner product’s iso-similarity planes with points on the unit sphere, yielding circles. These combine with the normalization lines to give iso-similarity contours that are cone-shaped. More generally, the contours of iso-similarity in $n$-space take the shape of $n-1$ dimensional “hyper-cones”.⁹ The first thing to note about these

---

⁷The cosine coefficient is one measure used in the SMART information retrieval system [19].

⁴This component might be justified under the assumption that the length of an object's vector representation is an irrelevant, perhaps misleading, feature that is best factored out of the similarity judgement. We might also assert that this component helps to equalize for the cost of intra-document retrieval. It might be reasonably assumed, for example, that the length of an object’s vector representation correlates with the length of the object (e.g., document) and that this length, in turn, is predictive of the costs associated with the extraction of a desired piece of information from the object.

⁸Under the assumption that all term weights are non-negative, points would be restricted to the positive quarter circle, or more generally the positive orthant of the hypersphere.

⁹The reader might be wondering why this circuitous path was taken. After all, the cosine measure might be simply taken as assigning equal similarity to points that have the same cosine with the query, and hence the same angle with the query. The point here was to understand, algebraically and geometrically, the separate normalization and comparison aspects of the measure, in part to introduce this sort of analysis-by-decomposition (it will be useful again later), and in part to emphasize normalization as a critical aspect of the cosine measure.
FIG. 4. The derivation of iso-similarity contours for the cosine measure: a. All points on the same radiating line normalize to the same unit vector (of L2-length = 1) and must be iso-similar, b. Unit vectors are iso-similar if they fall on the same inner product contour, c. The resulting iso-similarity contours are the pairs of radiating lines spaced according to the intersections in b.

Contours is how visually different they are from those of the inner product measure. There are corresponding differences in how these two measures behave in information retrieval, as we will see by describing the cosine coefficient with respect to the similarity measure attributes that were previously used to discuss the inner product:

- **Angle monotonicity**—As an inspection of Figure 4c illustrates, a decrease in the angle of separation between a query and an object vector is accompanied by an increase in the cosine similarity rating, so the cosine measure is angle monotone. In fact, the cosine measure illustrates a much stronger version of angle monotonicity than that we saw in the analysis of the inner product. Here angle monotonicity holds regardless of vector length. That is, in so far as direction corresponds to topic, we can say that topic dominates the cosine measure.

\[Q = (0.5, 1.0)\]

\[Q = (0.5, 1.0)\]

\[\text{We also note that the spacing of the cosine measure is not uniform, i.e., it is somewhat less sensitive at small angles. The import of this feature depends upon the actual use of the similarity values. The spacing is, of course, irrelevant if we simply wish to rank order the database objects according to their similarity to the query.}\]
• Radial monotonicity—The cosine measure, by virtue of its normalization, is in fact radial-constant. Changes in the length of an object vector have absolutely no effect upon its similarity rating. Thus direction (or semantic topic) not only dominates the measure, it does so completely.

• Component-wise monotonicity—The addition of a non-negative vector to an object’s vector improves the object’s cosine similarity rating only if this addition decreases the angle of separation between query and object vectors. When the addition of a non-negative vector brings about an increase in the object vector’s angle of separation, then the object’s cosine similarity rating decreases.

• Unbounded single-component influence—The cosine measure gives the highest similarity rating to objects whose term weight proportions most closely match those of the query. An extremely high term weight in an object’s representation will only tend to increase the object vector’s angle of separation from the query and so lower its similarity rating. Hence, the cosine measure is not susceptible to any inflation from a single component.

• Boundedness of similarity values—For non-negative object and query vectors, the cosine similarity values range from zero to one. A related property of this is that, unlike the inner product, there can exist maximally similar objects—namely those lying anywhere on the ray from the origin through the query vector.

This attribute characterization of the cosine measure describes a situation wherein an object’s similarity rating is solely determined by its topic as this is expressed through within-object term relationships. The cosine measure, therefore, avoids the possible distortions of the inner product measure resulting from an unbounded influence of individual vector components. But this avoidance is purchased at the price of ignoring potentially important between-object term weight relationships which may express the relative topical intensity of different objects. Beyond this, the cosine measure effectively imposes a “representational penalty”; an object is penalized for term weightings that are not represented (i.e., are zero-valued) in the query since these increase the object vector’s angle of separation from the query vector.

The Pseudo-Cosine Measure

In the cosine measure, normalization is based upon the L2 (Euclidean) lengths of query and object vectors. In this section we introduced a pseudo-cosine measure in which normalization is based, instead, upon the L1 or city-block lengths of query and object vectors. This measure is both instructive at this point and will be useful later when we consider the Dice similarity measure. The formula for the pseudo-cosine is:

\[
\frac{\sum_{i=1}^{n} W_{Oi} \ast W_{Oi}}{\left( \sum_{i=1}^{n} W_{Oi} \right) \ast \left( \sum_{j=1}^{n} W_{Oj} \right)}
\]

Algebraically, the pseudo-cosine differs from the cosine only in the way it measures vector length for the purposes of normalization. The L1 length is simply the arithmetic sum of the vector’s components. This contrasts with the square-root-of-sum-of-squares used to calculate the L2 lengths of the cosine measure. Like the cosine measure, the pseudo-cosine may be reconceptualized as the inner product of normalized vectors. In information retrieval this normalization is more typically thought of as part of a storage-time term weighting algorithm rather than as a retrieval-time component of a similarity measure (see 1, 12).

What do the iso-similarity contours of the pseudo-cosine look like? Following the analysis of the true cosine measure, we will first consider the effect of the normalization and then of the comparison. The normalization again takes all points on the same ray from the origin and maps them onto a single point on that ray at L1 distance 1 from the origin (Figure 5a). This means, again, that all points on any such radiating line will have the same similarity to the query, regardless of the comparison operation. If the comparison operation is an inner product, as it is here, we must look at how the inner product contours intersect the set of normalized points to find which subsets of the normalized points are themselves iso-similar (see Figure 5b). This time, however, the normalized points do not all fall on the unit circle but on the line \(x + y = 1\), and the result is quite different from the true cosine. Intersecting the parallel inner product contours with this line yields intersection points evenly spaced along the line and increasing monotonically in one direction. The direction of increase depends upon which side of the diagonal (the ones vector) the query is on, i.e., whether the query is closer to the \(x\)- or the \(y\)-axis.

Putting the normalization lines back in the picture (Figure 5c), we arrive at a complete 2-dimensional iso-similarity picture that on first glance may look like the cosine picture. In both Figures 4c and 5c, regions of comparable similarity (to within 0.025) are conical shaped (as indicated by the presence or absence of shading). This resemblance comes from the fact that both have a multiplicative normalization (1/length in some sense). Beyond this, the two measures differ quite a bit: In the cosine case, the similarity values are symmetric around the query and reach maximum value in the query’s direction. In the pseudo-cosine case, they are not symmetric and are in fact highest, not at the query itself, but at the coordinate axis that is closest to the query — call it the preferred direction. This preferred direction corresponds to the most heavily weighted term in the query. The result is that the most highly rated objects will be those that have only a single non-zero term corresponding to the most heavily weighted term in the query. The situation is even more bizarre if terms in the query are equally weighted, so that the query falls on the diagonal. In this case, the pseudo-cosine measure degenerates and all objects in the space receive the same similarity rating, regardless of their representation. The 2-space visualization of this situation is that iso-similarity contours of the inner product parallel the \(x + y = 1\) line so that one of these includes all normalized points of the \(x + y = 1\) line.
FIG. 5. The derivation of iso-similarity contours for the pseudo-cosine measure: a. All points on the same radiating line normalize to the same vector of $L_1$-length $= 1$ and must be iso-similar. b. The similarity spacing of $L_1$-length $= 1$ vectors are determined by the contours of the inner product. c. This spacing is then extended back into the original space along radiating lines to determine the actual iso-similarity contours.

We now try to generalize, to an arbitrary $n$-space, our visualization of the pseudo-cosine. In 3-space the normalized points fall on the $x + y + z = 1$ plane, which when clipped to be inside the non-negative part of the space is an equilateral triangle whose corners are at the points $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$. The planes generated by the inner product measure intersect with this plane to produce a pattern of parallel lines. The extension of these lines into 3-space along the rays of $L_1$ normalization produces a fan of iso-similarity planes that are flat in contrast to the iso-similarity cones produced by the true cosine. In agreement with the 2-dimensional example, the most highly rated objects will be those that normalize onto the corner of the equilateral triangle corresponding to the most heavily weighted term in the query.

In a general $n$-space, the pseudo-cosine will produce $n-1$ dimensional hyper-planes of iso-similarity with a ray of maximal similarity corresponding to a preferred direction that is determined by the most heavily weighted term in the query. This situation sharply contrasts with the cosine case in which $n-1$ dimensional hyper-cones of iso-similarity are produced and a ray of maximal similarity corresponds...
to the direction of the query itself. From another perspective, the pseudo-cosine measure can be seen to severely limit the expressive power of queries since, in an \( n \)-space, it allows them to express only \( n \) preferred directions (in contrast to the infinite number of preferred directions allowed by the true cosine). It is also true in a general \( n \)-space that a query lying along the diagonal produces a degenerate case of the pseudo-cosine measure in which all objects receive a similarity rating of \( 1/n \), regardless of their location in space.

We now summarize the pseudo-cosine coefficient with respect to the similarity measure attributes that were previously used to discuss the cosine and inner product:

- **Angle monotonicity**—The pseudo-cosine measure of similarity increases with decreases in the angle of separation between an object vector and the axis representing the dominant term of the query. This measure does NOT increase with decreases in the angle of separation between object and query vectors.

- **Radial monotonicity**—Because, like the cosine, it has a multiplicative normalization, increases in the length of an object vector have no effect upon the pseudo-cosine measure of similarity.

- **Component-wise monotonicity**—Component-wise increases tend to improve an object's pseudo-cosine measure of similarity only if these increases decrease the object vector's angle of separation from the axis representing the dominant term of the query. Component-wise increases that increase this angle of separation will bring about a decrease in the object's similarity rating. An object is generally penalized, therefore, for its representational richness.

- **Unbounded single-component influence**—The pseudo-cosine is susceptible to some inflation from the most heavily weighted term, but not to an unbounded degree and not, in general, from any single component.

- **Boundedness of similarity values**—For non-negative object and query vectors, the pseudo-cosine similarity values are also bounded, but the range depends on the position of the query. If the query is itself on an axis, similarity values range from 0 (at any other axis) to 1 (at that axis). If the query is on the diagonal then the measure degenerates and ALL objects have the similarity value \( 1/n \) (in \( n \)-space), regardless of their location in space.

The examination of the pseudo-cosine further illustrates an analysis that works by decomposing a similarity measure into separate components of normalization and comparison. The pseudo-cosine measure is rather strange. As a consequence of its non-monotonic behavior, it exhibits insensitivities to within-object, within-query and between-object term weight relationships. Beyond this, the range of pseudo-cosine similarity values is query-dependent and its behavior degenerates for certain values of the query. These features would make it of questionable use in information retrieval. The important caveat, though, is to remember that this pseudo-cosine is exactly what results if the inner product is applied to objects that have been normalized by the arithmetic sum of their term weights\(^2\)—a combination that might otherwise be tempting. As we shall also shortly see, the behavior of the pseudo-cosine also arises in a limiting case of the Dice measure.

### The Dice Measure

We turn now to a consideration of the Dice measure of similarity [4] whose formula is the following:

\[
2 \times \frac{\sum_{j=1}^{n} (W_{qj} \times W_{oj})}{\sum_{i=1}^{n} W_{qi} + \sum_{j=1}^{n} W_{oj}}
\]

The Dice measure, then, is obtained by taking the inner product of the query and object vectors, multiplying this number by two, and then dividing the result by the sum of the L1 (city-block metric) lengths of these two vectors. Since the denominator of this formula is additive, not multiplicative, we cannot break this down directly into separate operations of normalization and comparison. In fact, the overall form of this equation (and hence also the pattern of iso-similarity contours) changes drastically depending upon the relative L1 lengths of the query and object vectors.

We can gain an understanding of this dependency by considering the two extremes of relative L1 length. For a fixed object set, we can make the query vector arbitrarily long in relation to the lengths of the object vectors so that the effect of object length in the denominator becomes arbitrarily small. In this extreme case, we have a Dice\(_1\) measure that simply equals the inner product times a constant. The nature of this measure's similarity contours are thus identical to those of the inner product (see Figure 2). For a fixed object set, we can also make the query vector arbitrarily short in relation to the lengths of the object vectors so that the effect of query length in the denominator becomes arbitrarily small. In this case the formula becomes a Dice\(_2\) measure that is simply the pseudo-cosine times a constant. The nature of this measure's similarity contours are thus identical to those of the pseudo-cosine (see Figure 5).

Stepping back from the limiting cases of the Dice, and Dice\(_2\) measures, it is apparent that the actual performance of the Dice measure will generally fall somewhere between these extremes. In fact this means that if we draw a query vector (see Figure 6) at some intermediate length, the area near the origin will have contours perpendicular to the query like those of the inner product. In the area far outside the query the rays will approach asymptotes that radiate from the origin like those of the pseudo-cosine. (Again, as with the figures illustrating the cosine and pseudo-cosine, all vectors within a particular shaded or unshaded region have roughly the same similarity rating).

\(^2\)Note again that this normalization might have occurred prior to the retrieval-time application of the inner product. (It might, for example, have occurred during the initial selection of an object's term weights).
The particular properties of the measure can be considered by referring to properties of the limiting cases. We need not go through the list again. We simply note that, at best, the Dice measure will approximate the functionality of the inner product. In the worst case, the Dice measure will exhibit the insensitivities of the pseudo-cosine. Actual performance of the Dice measure will fall somewhere between these extremes, depending upon the relative lengths of query and object vectors.

Measures of Product-Moment Correlation and Covariance

We deal with covariance and correlation together because of their close relationship to one another and to the inner product and cosine measures. The formula for the covariance is:

$$\sum_{i=1}^{n} (W_{qi} - \bar{W}_q) \times (W_{oi} - \bar{W}_o)$$

This measure essentially applies the inner product to new vectors $Q'$ and $O'$. A given component of $Q'$ ($O'$) is formed by subtracting the corresponding component in the original vector $Q$ ($O$) by the average value of a term weight in the original vector.

The formula for the product-moment correlation is:

$$\frac{\sum_{i=1}^{n} (W_{qi} - \bar{W}_q) \times (W_{oi} - \bar{W}_o)}{\sqrt{\sum_{k=1}^{n} (W_{qk} - \bar{W}_q)^2} \times \sqrt{\sum_{j=1}^{n} (W_{oj} - \bar{W}_o)^2}}$$

This measure essentially applies the cosine coefficient to the new vectors $Q'$ and $O'$ (as described above). Consequently, we might say that the covariance measure is to the inner product measure what the correlation measure is to the cosine measure.

Common to both the covariance and the correlation measures is a moment normalization of query and object vectors which is accomplished by subtracting each component of a given vector by the mean value of the vector's components. Figure 7 depicts the effects of this normalization in 2-space. All vectors falling on the same line parallel to the diagonal will be transformed by this normalization into the same vector whose components sum to zero. The result in 2-space is that all normalized query and object vectors lie upon the negative diagonal line, $x + y = 0$. Since any subsequent comparison is mediated only by the normalized points, all points falling on the same normalization line will have the same similarity to the query. Generalizing to $n$-space, normalized points will all fall on the $n-1$ dimensional hyperplane that is perpendicular to the diagonal and that passes through the origin.

With moment normalization we lose a degree of freedom in the representational richness (or expressive power) of query and object vectors. Alternatively, we are discarding information from the representations of query and object vectors. As with the cosine's L2 length normalization, the notion might be that this information is irrelevant or misleading. The moment normalization leaves us with a vector whose components reflect deviations of corresponding components in the original vector from the mean of the original vector. Implicit in its use is a notion that these deviations, and not the absolute values in the original vectors, are pre-
dictive of an object's relevance to a query.\textsuperscript{13} It should be emphasized, however, that the moment normalization distorts the within-object term weight relationships of the original vectors. In particular, information regarding the ratios of term weights in the original object vector is lost. The moment normalization effects an even more severe distortion of between-object term weight relationships. There is really nothing that can be reliably inferred from between-object term weight relationships in a set of moment normalized vectors since even the ordering of a term's weights may change depending upon the means of the original object vectors.

Moment normalization produce anomalies in the 2-space depictions of both the covariance and correlation measures. We note, for example, that a normalized query vector $Q'$ can assume only one of two directions corresponding to the two directions of the $x + y = 0$ negative diagonal. These anomalies make it more difficult to arrive at the proper n-space generalizations than it was in the case of previously discussed measures.

To characterize the covariance measure we can follow the same approach we have applied to the cosine and pseudo-cosine. In doing so we move from a consideration of moment normalization to a consideration of the inner product comparison operation as it is applied to the space of normalized vectors. However, since any normalized query vector $Q'$ must lie on the $x + y = 0$ line in 2-space, the pattern of iso-similarity contours is the same, regardless of the original query, and this pattern is indistinguishable from that of the moment normalization operation. Only the ordering of similarity values associated with a contour line will vary, i.e., values can either increase or decrease as we move along the $x + y = 0$ line depending upon whether the query vector normalizes to the first or the third quadrant.

Generalizing to $n$-space, we have already noted that the moment normalization induces a projection of the object and query vectors of an $n$-space onto an $n-1$ dimensional subspace that is perpendicular to the diagonal and that passes through the origin. The normalized query vector $Q'$ in this subspace establishes a "preferred direction". If the angle separating an object $O$ from this preferred direction is less than 90 degrees, then object $O$ will receive a positive similarity rating; if its angle of separation is more than 90 degrees, its similarity rating will be negative. In $n$-space, the covariance measure produces, as contours of iso-similarity, $n-1$ dimensional hyper-planes lying perpendicular to the preferred direction established by the transformed query vector $Q'$ (and, consequently, lying parallel to the diagonal).

We can now describe the covariance measure with respect to previously discussed similarity measure attributes:

- **Angle monotonicity**—With respect to the original query vector, the covariance measure is not angle monotone. That is, holding L2 (Euclidean) length constant, if we change a vector's direction so that its angle of separation from the original query vector decreases, we do not necessarily increase its similarity rating. We may, in fact, decrease its rating if this rotation increases the vector's angle of separation from the preferred direction established by the transformed query $Q'$. The covariance measure is, however, angle monotone with respect to the transformed query $Q'$.

- **Radial monotonicity**—If, for a given query, an object vector has a positive similarity rating, then increases in its length (keeping its direction constant) will effect an increase in this similarity rating; if the object vector has a negative rating, then increases in its length will diminish its similarity rating still further. The covariance measure, therefore, is not radial monotone.

- **Component-wise monotonicity**—The covariance measure is not component-wise monotone. The transformation of object vector $O$ through the addition of a non-negative vector $C$ will increase its similarity rating only if the direction of vector $C$ is "close" to that of the preferred direction (i.e., if its angle of separation from this direction is less than 90 degrees).

- **Unbounded single-component influence**—In spite of the moment normalization, the covariance measure, like the inner product measure, can produce distortions resulting from its absolute interpretation of term weights. For example, given a three-space query vector $Q = (9, 6, 0)$ the covariance measure can be made to give an arbitrarily large similarity rating to an object vector $O = (0, w, 0)$ by simply increasing the magnitude of $w$.

- **Boundedness of similarity values**—Even for non-negative object and query vectors, the covariance similarity values can range unboundedly, from positive to negative infinity.

We turn now to a consideration of the correlation similarity measure. In this case, there are two normalizations. First, a vector is moment normalized to additively remove its mean from each of its components. The resulting vector is then normalized by L2 (Euclidean) length. Two normalizations applied to a 2-space of points results in a situation still more anomalous than that we observed for the covariance measure. As illustrated in Figure 8, the correlation measure produces exactly two regions of iso-similarity. One region is bounded by the x-axis and the diagonal while the other region is bounded by the y-axis and the diagonal. If an object vector lies in the same region as the query vector, it receives a maximal rating of 1.0; if it lies in the opposite region, it receives the minimum rating of 1.0. (If it is coincident with the diagonal, its rating is 0).

The situation is slightly more interesting in 3-space. The moment normalization maps all points on a given line that parallels the diagonal onto the same point in the plane that is perpendicular to the diagonal and that passes through the origin (i.e., the plane whose points satisfy the equation, $x + y + z = 0$). The cosine-like L2 normalization further maps all points on a ray in the $x + y + z = 0$ plane onto a single point in the unit circle that is embedded in that
is-o-similarity regions in two space for the correlation similarity measure.

Finally, the inner product comparison establishes an iso-similarity pairing between the points of this unit circle (i.e., points with the same angle of separation from the normalized query \( Q' \) are given the same similarity value). Reversing this process, we obtain iso-similarity contours in the original unnormalized space of vectors that take the form of pairs of half-planes. The members of a pair meet at and emanate from the origin along the rays of \( x + y + z = 0 \) plane unit circle and along lines of moment normalization. The effect is that contours of iso-similarity have a wedge-like appearance. In \( n \)-space, the correlation measure's contours of iso-similarity will take the form of \( n-1 \) dimensional wedges lying parallel to the diagonal and at varying angles of separation from the preferred direction established by the moment normalized query vector \( Q' \).

We can now describe the correlation coefficient with respect to previously discussed attributes of a similarity measure:

- **Angle monotonicity**—The correlation measure is not angle monotone with respect to the original query vector \( Q \). Moreover, in contrast to the covariance measure, it is also not angle monotone with respect to the moment normalized query vector \( Q' \). We can, however, introduce another notion of angle monotonicity that the correlation measure does satisfy. Working from the normalized query vector \( Q' \), we can reverse the \( L_2 \) and moment normalizations of the correlation measure to construct a 2-dimensional vector subspace—call it the \( Q \)-plane—which contains both \( Q \) and \( Q' \), together with all linear combinations of these two vectors. Any vector \( O \) then has an angle of separation from the \( Q \)-plane that is defined to be the minimum of its angles of separation from various vectors of the \( Q \)-plane. It turns out that the correlation measure satisfies the following version of angle monotonicity: For any two objects, the correlation measure gives the higher similarity rating to the vector with the smaller angle of separation from the \( Q \)-plane—the two-dimensional vector subspace containing the original query and moment normalized queries.

- **Radial monotonicity**—The correlation measure is not radial monotone—increases in the length of an object vector have no effect on its similarity rating.

- **Component-wise monotonicity** More generally, the correlation measure is not component-wise monotone. Component-wise modifications improve an object's similarity rating only when they decrease its angle of separation from the \( Q \)-plane.

- **Unbounded single-component influence**—The correlation measure is not bounded by the numbers \(-1\) and \(+1\).

In summary, the covariance and correlation measures seem to be subject to all of the potential limitations of their respective inner product and cosine counterparts. Additional limitations may result from the use of moment normalization and the resulting loss of potentially useful information.

The Overlap Measure

We turn now to a consideration of the overlap coefficient:

\[
\frac{\sum_{i=1}^{n} \min_{j}(W_{Qi}, W_{Oj})}{\min \left( \sum_{i=1}^{n} W_{Qi}, \sum_{j=1}^{n} W_{Oj} \right)}
\]

Of the similarity measures considered in this paper, the overlap coefficient is unique for the absence of an inner product component to its formula. The numerator of the overlap coefficient can be viewed as an alternate comparison operation that also computes a form of vector intersection. In the boolean case (where term weights can assume only the values of zero and one), the inner product and the numerator of the overlap coefficient produce identical values.

The form of the denominator of the overlap measure is also unique among those we consider in this article. The value actually used in the denominator will depend upon the relative \( L_1 \) (city-block) lengths of query and object vectors. In other words, like the dice metric, the form of the contours will vary at different distances from the origin. The overlap denominator acts to normalize the vector intersection measure of the numerator so that similarity values lie within a range of \( 0 \) to \(+1\). We will see that the nature of iso-similarity curves that result from the overlap measure's choice of com-

\*The overlap measure is used in the Smart retrieval system [19].
Comparison and normalizing components give a very strange interpretation to the database set of object vectors.

Representative contours of iso-similarity (and, perhaps more appropriately, regions of iso-similarity) are presented in Figure 9. A given query produces two regions or “boxes” in which object vectors are given a maximal similarity rating of 1 (the highest possible score). Generalizing to n-space, the overlap measure, for any given query vector, produces two rectilinearly bounded regions within which object vectors are given a maximal similarity rating of 1. One box of maximum similarity is anchored by the origin; object vectors lying within this box are everywhere, or component-wise, smaller than the query vector. The other box is anchored by the query vector; object vectors within this box are larger, component-wise, than the query vector. We thus have a very curious situation in which object vectors that have all components either very small or very large will get a maximal similarity rating. The measure is thus comparatively insensitive to both within- and between-object term weight relationships.

Object vectors in the remaining regions of partial overlap are neither larger nor smaller, component-wise, than the query vector. In a general n-space we will find 2^n regions of partial overlap, each with dimensionality equal to n. The regions of partial overlap in Figure 9 are partitioned into subregions by a line that lies perpendicular to the diagonal and includes the query vector. In a general n-space, regions of partial overlap will each be partitioned by an n-1 dimensional hyper-plane that lies perpendicular to the diagonal and includes the query vector. A vector in such a hyper-plane has an L1 length (the sum of its components) equal to that of the query vector. For a given object vector lying on one side of this hyper-plane, the overlap measure normalizes through division by the L1 length of the query vector; for a given object vector lying on the other side of this hyper-plane, the overlap measure normalizes through division by the L1 length of the object vector.

We also note that the similarity ratings of the object vectors in a region of partial overlap will all be above a region-specific minimum value. These values are determined by the manner in which the region is bounded by component values of the query vector. The absolute minimum similarity across all regions equals the smallest component of the query vector divided by the L1 length of the query vector.

We can now analyze the overlap measure with respect to the same set of attributes we have used to assess other similarity measures.

- **Angle monotonicity**—The overlap measure is angle constant within a query vector’s two regions of maximal similarity; it is angle monotone only within regions of partial overlap.
- **Radial monotonicity**—It is with respect to this attribute that the overlap measure is most erratic. The measure is radial monotone only in subregions of partial overlap corresponding to situations in which the overlap measure is normalized by the L1 length of the query vector (subregions above the line lying perpendicular to the diagonal in Figure 9). Increases in vector length in these regions will eventually push a vector into the upper region of maximal similarity. Within both regions of maximal similarity, the overlap measure is radial constant. However, in the lower region of maximal similarity, increases in vector length will eventually push an object vector into a subregion of partial overlap in which the overlap measure is normalized by the L1 length of the object vector (unless the object vector is coincident with the query vector). These subregions are below the line lying perpendicular to the diagonal in Figure 9. Finally, within these subregions of partial overlap, the overlap measure is monotonically decreasing with increases in vector length!
- **Component-wise monotonicity**—The tortuous situation described above with respect to the attribute of radial monotonicity applies here as well. The overlap similarity rating may increase, decrease, or remain the same with component-wise increases depending upon the region that the object vector is currently located in and the region (or subregion) towards which a component-wise increase directs the modified object vector.
- **Unbounded single-component influence**—Under the overlap measure, components of the query vector establish an upper limit on the effect that any given term dimension can exert on the measurement of similarity. The overlap measure is thus resistant to the unbounded influence of a single component.
- **Boundedness of similarity values**—Overlap similarity ratings range from the minimum established by the minimal component of the query vector (as this was previously discussed) up to a maximum of one.

A primary concern with the overlap measure relates to its creation, for any given query, of two distinct regions of maximal similarity. In doing so, the measure favors object vectors that are, component-wise, either very short or very long. Additional concerns center upon the measure’s very
erratic patterns of non-monotone behavior. The overlap measure exhibits a general insensitivity to both within- and between-object term weight relationships.

The Spreading Activation Measure

The final measure we consider here, the spreading activation measure, is chosen as a representative of a class of context-dependent similarity measures in which the similarity value of a given object depends upon the collection of objects as a whole. The measure is motivated by proposed mechanisms of human memory retrieval [5] [6] [7] [8] [9], and the formula for this measure is best introduced by sketching a mechanism that can be used to implement the formula in a computational setting [20] [21]. In the mechanism depicted in Figure 10, the query, objects, and terms are represented as nodes interconnected by a network of bi-directional links. Note that links only connect terms to objects or queries, and not terms to terms, or objects to each other or to the query. Thus there is a link in the network for every cell in a term-by-object weight matrix (such as those of Tables 1 and 2) and we associate the weight in a cell with the corresponding link. (For graphical convenience, one might leave out links with weight zero.)

The retrieval process begins when an amount of activation $A$ (perhaps equal to 1.0) is deposited at the top node in Figure 10, where this node represents the query. The activation then “spreads” outward on all the links attached to the query. Obeying a kind of conservation of activation principle, the total activation is divided up in amounts proportional to the link weights, and in these proportions activates each of the query’s associated terms (represented by the second tier of nodes in Figure 10). Thus the activation $a_i$, received by a given term $i$ is:

$$a_i = A \cdot \frac{W_{Qi}}{\sum_{k=1}^{n} W_{Qk}}$$

The denominator in this expression is the sum of all the query’s term weights.

The process continues when the activation received by term $i$ is further partitioned among its associated object nodes according to the relative weights of their connecting links. The activation $a_o$ received by a given object $O$ from term $t$ in this manner is computed as follows:

$$a_o = a_i \cdot \frac{W_{Qi}}{\sum_{j=1}^{m} W_{Qj}}$$

The denominator in this expression is the sum of all object weights for term $t$.

The activation that a given object $O$ receives in this manner from several different query terms will sum together and this sum represents a spreading activation measure of the object’s similarity to the query. Equivalently, in the notation used to represent conventional vector similarity measures, the spreading activation measure of an object’s similarity is computed as follows:

$$A = \left( \sum_{i=1}^{n} \frac{W_{Qi} \cdot W_{QO}}{\left( \sum_{k=1}^{n} W_{Qk} \right) \cdot \left( \sum_{j=1}^{m} W_{Qj} \right)} \right)$$

It should be emphasized that this is a limited version of the spreading activation process. It is certainly possible, for example, to allow activation to continue to reverberate through the network beyond the two stages described here.

The spreading activation formula has an appearance much like those of the formulas associated with other similarity measures. In common with all other measures discussed in this article except the overlap coefficient, the spreading activation formula has a numerator that is essentially an inner product measure of query and object vector overlap. Part of the denominator is simply the L1 length of the query vector and we might view this as an attempt to normalize the length of the query vector. As a result of this normalization all queries in the same direction have identical contours of iso-similarity.

It is in the second factor of the denominator that the spreading activation formula differs markedly from those associated with other similarity measures. This factor has the effect of normalizing each term weight in an object’s representation. However, for a given term weight, this is
done not with respect to other term weights of the given object but with respect to the weights of the term across all other objects. Normalization is thus term-relative rather than object-relative (as in the case of the cosine measure, for example).

Sparck-Jones [12] reviews a collection of empirical findings demonstrating the advantages of term-relative normalization and also demonstrating the superiority of this kind of normalization to an object-relative normalization. There are several approaches to the term-relative normalization of weights (see [4] pp. 204–207) [11] [2]. These methods are typically applied at the time of the initial selection of an object’s term weights and they effect a permanent modification that can distort within-object term weight relationships. Moreover, the updating of term weights can be computationally very expensive even though such an updating may often be necessary in a continually changing collection if weights are to remain current. By contrast, the spreading activation mechanism is an implementation of term normalization that simultaneously realizes the attributes of database currency and interpretability along with the attribute of computational efficiency (see [20], [21]).

Under the spreading activation measure it might be said that an object is competing, through its term weights, with other objects in the database for a limited supply of activation. Because of the measure’s term-relative normalization, an object’s similarity rating is dependent upon a context established by other objects in the database. This context-dependent normalization of term weights makes it more difficult to give a geometric characterization to the iso-similarity contours of the spreading activation measure since contours will depend not only on the query vector but also on the overall distribution of object vectors. The effect of term normalization is, however, straightforward, and may be conceived of in two different ways: either as readjusting the object vectors of the collection or as readjusting the query vector.

From the first perspective, term-relative normalization has the effect of contracting the normalized object’s length along all term dimensions. But, as Figure 11 demonstrates, these contractions are relatively larger along dimensions whose corresponding terms are in frequent use across database objects. The highest value a normalized object vector can achieve on any given term dimension is one. And in general the values will be much smaller than one, having an average of \(1/m\), where \(m\) is the number of objects in the space. In 2-space, all normalized object vectors will fall within a unit square bounded by the \(x\)-axis, the \(y\)-axis, and the lines \(x = 1\) and \(y = 1\). In \(n\)-space, all normalized object vectors will fall within a unit hyper-cube that is anchored at the origin.

Once this normalization of objects is complete, all that remains is to apply the spreading activation comparison operation—which is simply the inner product applied between the original query vector and a normalized object vector. In the term-normalized space, therefore, contours of iso-similarity are orthogonal to the original query vector. Note that when the inner product is applied to objects in the original unnormalized space (Figure 11a) object \(A\) receives the lowest similarity rating (among the four objects represented), whereas its counterpart in the normalized space, object \(A'\), receives the second highest rating. This occurs because object \(A\) has large weighting to the relatively more discriminating term corresponding to the \(x\)-axis.

However, the contours in Figure 11b must be interpreted with caution, since only some of our previous uses of the iso-similarity contours are still valid. It is still true that two existing objects can be compared using the contours. However, in contrast to the situation for other similarity measures considered in this article, term-normalization makes it impossible to see simply from the contours how to change existing objects or how to introduce new objects to...
get desired results. This is because the change of an old object vector or the introduction of a new object vector forces a change in the normalized vectors of all other objects in the database which, in turn, invalidates existing iso-similarity contours.

We gain a slightly different geometric characterization of the spreading activation measure if we apply its term-relative normalizations to the query rather than to the objects (see Figure 12). This has the effect of creating a new normalized query vector $Q'$. If patterns of database usage are roughly the same across terms in the query, then the direction of $Q'$ will be roughly the same as the direction of the original query vector $Q$. Under other circumstances, however, vector $Q'$ will point relatively more strongly in the directions corresponding to less frequently used (and hence more discriminating) terms. Again, once the normalizations are complete, all that remains to assess an object $O$'s spreading activation similarity is to find its inner product with the normalized query vector $Q'$. Consequently, iso-similarity contours with respect to vector $Q'$ are identical to those of the inner product. In Figure 12, as in Figure 11, one effect of the term-relative normalization is to move object $A$ from last place to second place with respect to its similarity rating.

Again, however, the context-dependent nature of spreading activation normalization denies us the same freedom of interpretation that we enjoyed with other measures considered in this article. If two object vectors happen to lie along the same line of iso-similarity then we can conclude that the corresponding objects have identical measures of spreading activation similarity. On the other hand, we cannot arbitrarily change an existing object vector or add a new object vector to this vector space and infer much about its similarity rating on the basis of existing iso-similarity contours. This is because the placement of the new object effects a change in database term usage patterns. This change, in turn, effects a change in the direction of the normalized query vector $Q'$ which then invalidates all existing iso-similarity contours.

One advantage of this second characterization is that we now have the iso-similarity contours plotted in the original object space. With respect to vector $Q'$, these contours are exactly like those of the inner product. As a result, the contours are linear, equally spaced, and they are perpendicular to $Q'$. They are NOT, in general, perpendicular to the original query vector $Q$. With this understanding of the geometry of the spreading activation measure we can now describe the measure with respect to the set of attributes we have used to assess other similarity measures:

- **Angle monotonicity**—An object's measure of spreading activation similarity increases with decreases in the angle of its vector's separation from the normalized query $Q'$. However, similarity does not necessarily increase with decreases in the angle separating the object vector from the original query vector $Q$. Similarity rating may actually decrease, if the change in object vector direction reflects a shift in weight from a less frequently used term to a more frequently used term.

- **Radial monotonicity**—With the spreading activation measure, as with the inner product, an object's similarity rating increases or remains the same when the length of the object's vector is increased.

- **Component-wise monotonicity**—Again in agreement with the inner product measure, an object's similarity rating increases or remains the same with any increase in term weights; an object's similarity rating never decreases. In general, increases in an object's term weights enable it to more effectively compete with other objects for a limited supply of activation.
• **Unbounded single-component influence**—The term-relative normalization of weights limits the effect that a single term weight in an object's representation can have on its similarity rating. With reference to Table 3, for example, it is not possible to give object \( O_2 \) an arbitrarily high similarity rating through manipulations in \( w \). Since the corresponding query term weight is 1, out of a sum total of 14 in query term weights, object \( O_2 \) will never receive more than about 1/14 of the query's activation. More generally, the amount of activation that an object will receive due to its weighting to a given term \( i \) is bounded by the term's weight in the query vector relative to the weights of other terms in the query vector. Consequently, the spreading activation measure is resistant to an unbounded influence of single components.

• **Boundedness of similarity values**—The actions of the spreading activation measure are *activation preserving* such that the activation received by objects in the database from a given query will sum to a value that equals the amount of activation \( A \) that was initially deposited at the query node (see Figure 10). Consequently, similarity ratings are bounded from below by zero and from above by \( A \) (which can equal one if we like).

With reference to Figure 10, the actions of the spreading activation measure can be seen to produce a dual sensitivity to within-query and between-object relationships among term weights. In the first stage, activation is divided among query terms according to their relative weightings to reflect within-query considerations regarding the relative importance of query terms. In the second stage, the activation a given term receives in this way is then partitioned among its object links to reflect between-object considerations of relevance. Consequently, the spreading activation measure can be seen to exhibit properties of both the inner product and cosine similarity measures.

Finally, it should be noted that the spreading activation measure is unique among the measures considered in this article with respect to the asymmetrical nature of its similarity judgments. Under the spreading activation measure the order of comparison makes a difference. This is not a consideration in retrieval attempts since the order of comparison is fixed, i.e., the query is matched to database objects, not vice-versa. However, the order of comparison will make a difference when database objects are compared with each other under the spreading activation measure (as part of an attempt to construct a matrix of inter-object similarities, for example). When two objects \( O_1 \) and \( O_2 \) are compared, it will seldom be the case that \( s(O_1, O_2) = s(O_2, O_1) \). In general, a less richly indexed object characterized by a shorter vector will be rated more similar to a more richly indexed object characterized by a longer vector than vice versa.16

---

16It is interesting to note that this appears to be consistent with a variety of psychological studies on similarity judgments in people [22].

### 4. Discussion

**Differences Among Similarity Measures**

In this article we have adopted a geometric approach to the analysis of six conventional vector similarity measures and a seventh spreading activation measure. Using this geometric approach, the attempt is made to characterize a similarity measure by the nature of its iso-similarity contours. The characteristic contours of a measure help us to determine the measure's sensitivity to the various kinds of information (e.g., within- and between-object term weight relationships) that may be present in the representations of an object collection. This approach has helped to elucidate important differences among the similarity measures considered in this article.

The inner product is a basic computation of vector intersection or overlap; in a boolean case, where term weights are zero or one, the inner product can be used to compute the cardinality of set intersection. The iso-similarity contours of the inner product measure take the form of \( n-1 \) dimensional hyper-planes lying at right angles to the query vector. The inner product is angle, radial, and component-wise monotone. The inner product is thus sensitive to between-object and within-object differences in term weights (where these may express a topic and its intensity, respectively). Moreover, the inner product does not penalize objects for their representational richness, i.e., for term weightings with no corresponding representation in the query.

On the other hand, the inner product is susceptible to unbounded influence by a single component. The inner product will rate, as equally similar to a query, objects with widely different within-object relationships among their term weights. It is, in fact, possible to give an object an arbitrarily high similarity rating through manipulations in its weighting to a single term — even if this term is only weakly represented in the query. Related to this is the absence of an upper bound to the inner product scores. The problem of unbounded single-component influence arises from the inner product's absolute, rather than relative, interpretation of term weights. Relative interpretations are introduced through the application of various normalizing components to the computation of similarity. All other similarity measures considered in this article, except for the overlap measure, can be derived from the inner product measure through the introduction of normalizing components.

The cosine measure normalizes the inner product through its division by the \( L_2 \) (Euclidean) lengths of query and object vectors that are involved in a computation. In \( n \)-space, cosine contours of iso-similarity take the shape of \( n-1 \) dimensional "cones" surrounding the query vector. The cosine measure is angle monotone. Its normalizing components give it a resistance to the unbounded influence of a single component and also establish an upper-bound on the range of similarity values. On the other hand, the cosine measure is neither component-wise nor radial monotone. The length of an object's vector has no effect upon its similarity rating. Moreover, an object may be penalized for
subject vectors, we have a limiting Dice measure that behaves to the L2 normalization of the cosine measure and to under-

bining to between-object term weight relationships. The cosine measure is insensitive to between-object term weight relationships.

We introduced a pseudo-cosine measure as an alternative to the L2 normalization of the cosine measure and to understand the performance of the Dice measure in a limiting case. Algebraically, the pseudo-cosine formula differs from the cosine formula only in its use of an L1 (city-block) length normalization rather than an L2 normalization. In a general n-space, the pseudo-cosine produces contours of iso-similarity that are flat hyper-planes in contrast to the conical shapes produced by the true cosine. More importantly, these contours increase in value to a maximum corresponding not to the direction of the query vector but, instead, to the axis of the term most heavily weighted in the query. The pseudo-cosine thus inflates the importance of this term in its calculations. The pseudo-cosine is not angle monotone, radial monotone, or component-wise monotone. On the basis of these attributes, the pseudo-cosine can be seen to exhibit insensitivities—to within-object, within-query and between-object term weight relationships.

The behavior of the Dice measure critically depends upon the relative L1 lengths of query and object vectors. As the query vector becomes arbitrarily long in relation to the object vectors, we have a limiting Dice measure that behaves like the inner product measure. On the other hand, as the query vector becomes arbitrarily short in relation to the object vectors, we have a limiting Dice2 measure that behaves like the pseudo-cosine measure. For intermediate values of the query vector, the behavior of the dice measure is somewhere between the Dice1 and Dice2 extremes. At best, therefore, the Dice measure will approximate the functionality of the inner product; in the worst case, the Dice measure will exhibit the insensitivities of the pseudo-cosine.

Common to both the product-moment covariance and correlation measures is a moment normalization of query and object vectors which is accomplished by subtracting each component of a given vector by the mean value of the vector’s components. The covariance measure is just the inner product computation applied to these moment normalized vectors. Similarly, the correlation measure is just the cosine computation applied to these normalized vectors. It is not surprising, then, that covariance and correlation measures share some of the characteristics of their respective inner product and cosine measure counterparts.

The use of moment normalization in these measures introduces additional potential drawbacks. Geometrically, moment normalization can be seen to collapse all points of a given line lying parallel to the diagonal onto a single point that lies in the hyper-plane that is perpendicular to the diagonal and that passes through the origin. Moment normalization removes a degree of freedom from the expressive power of query and object vectors. More importantly, its application distorts potentially important term weight relationships—both within-object and between-object. Although covariance and correlation measures are useful statistical measures of inter-vector predictiveness (i.e., they measure the extent to which one vector can be generated from a single interval transformation of another vector), these measures appear to have considerably less utility as assessors of inter-vector relevance (i.e., relevance between query and object vectors).

The overlap coefficient is unique among the measures considered in this article for its lack of an inner product component. Vector intersection or overlap is instead computed through a summation of component-wise minimum values in object and query vectors. This computation is then normalized through division by the minimum between the L1 lengths of query and object vectors. The overlap measure gives a maximal similarity rating to objects whose vectors are either larger or smaller, component-wise, than the query vector. Geometrically, these appear in n-space as two rectangular bounded regions—one anchored by the origin and the other anchored by the query vector itself. Within these two regions, the overlap measure is angle, radial, and component-wise monotone constant. Outside of these regions, in regions of partial overlap, the measure produces patterns of similarity that are erratic and decidedly non-monotone. In general, the overlap measure exhibits potentially serious insensitivities to both between-object and within-object term weight relationships.

The spreading activation measure is unique among the similarity measures considered in this article for its use of a term-relative normalization as opposed to an object-relative normalization. This normalization has the effect of favoring query terms with higher discrimination value (as measured by frequency of database usage). This normalization also gives the spreading activation measure a resistance to the unbounded influence of single components that is not purchased at the cost of ignoring between-object term weight relationships (e.g., unlike the object-relative normalization of the cosine measure). Objects, moreover, are never penalized for their representational richness.

However, as a consequence of this term-relative normalization, the spreading activation measure is typically not angle monotone with respect to a given query. The term-relative normalization has the effect of rotating the query vector so that it points more strongly in directions corresponding to the more discriminating terms in the query’s representation. The spreading activation measure is angle monotone with respect to this rotated query vector. It is angle monotone with respect to the original query vector only when all of the query’s term components have equal frequencies of database usage.

The context-sensitive, term-relative normalization of the spreading activation measure can be regarded as an attempt to establish a balance between within-query considerations of term importance and between-object (or within-term) considerations of object relevance. In the first stage of activation spread, within-query considerations are expressed through the apportioning of activation among query terms according to their relative weights. In the second stage of activation spread, between-object considerations of rele-
Similarity Measure Extensions

between-object term weight relationships, has chiefly been measures that are amenable to a geometric analysis. How-

example, a similarity measure that is based upon (e.g., some its focus on a measure's sensitivity to within-object and

Vance are expressed, for a given query term, through an apportioning of activation among its object associates ac-

to a term-relative normalization of similarity ratings. Many questions arise concerning the optimality of the spreading activation's term relative normalization both as a means of computing a term's discrimination value and as a means of establishing a tradeoff between within-query considerations of term importance and between-object considerations of object relevance. However, the spreading activation measure offers a promising approach to the assessment of object relevance and it is clearly deserving of further study.

The spreading activation measure also illustrates one limitation of the geometric approach to the analysis of similarity measures. For a class of context-dependent similarity measures (e.g., measures that use term normalization). iso-

similarity contours must be interpreted with some care. With these measures we cannot, for example, infer much about the similarity rating of a new or modified object vector on the basis of existing iso-similarity contours. This is because the addition or modification of an object effects a change in database term usage patterns which, in turn, effects a change in iso-similarity contours. Nevertheless, the geometric approach still proves useful, as the discussion of the spreading activation measure illustrates. For example, the approach provides us with a way of visualizing the effects of term normalization.

Clearly, this article presents only a small sampling of the measures that are amenable to a geometric analysis. How-

ever, the application of the geometric approach to many other measures is fairly straightforward. Consider, for example, a similarity measure that is based upon (e.g., some additive or multiplicative inverse of) the Euclidean (L2) distance between query and object vectors. In two-space, it can be readily seen that the iso-similarity contours of such a measure are concentric circles centered about the query vector. Similarity measures that are, instead, based upon the city-block (L1) distance between query and object vectors produce iso-similarity contours that are diamond-shaped rather than circular. In turn, similarity measures that are based upon a general Minkowski metric, Ln, as n goes to infinity, produce iso-similarity contours that are increasingly square-shaped. Regardless of their shape (e.g., diamonds, circles, squares), distance-based similarity mea-

sure contours are "centered" about the query vector. As a consequence of this centering, these similarity measures are angle monotone and are not vulnerable to unbounded single-component influence. On the other hand, these mea-

sures are neither radial nor component-wise monotone.

Similarity Measure Extensions

The analysis of similarity measures in this article, with its focus on a measure's sensitivity to within-object and between-object term weight relationships, has chiefly been object-oriented. However, at the outset we noted that a database matrix of term weights could conceivably carry other information as well. In particular, the pattern of a term's usage may contain important information regarding the term's discrimination value and its similarity to other terms in the database.

Through its sensitivity to term discrimination value, a similarity measure can enhance the precision of a query by giving preferential treatment to the query's more discriminating terms. In this regard, the spreading activation measure is unique among the similarity measures we have considered in this article. Through its term-relative normalization factor, the spreading activation measure gives preferential treatment to query terms that are less frequently used in the database. However, as we have already noted, there are other ways to introduce a sensitivity to term discrimination value into the computations of a similarity measure (see [4] pp. 204–207 [1] [2]) and the efficacy of various approaches is clearly in need of further study.

Through its sensitivity to term similarities, a similarity measure can also enhance the recall rate of a query. In general, the recall rate of a query can be improved (without necessarily compromising its precision) through the intro-
duction of new terms that are judged to be synonymous or nearly synonymous to terms explicitly represented in the query. Term synonymy, in turn, can sometimes be reasonably inferred by using a similarity measure to order terms of the database according to their rated similarity to query terms. Under this approach we essentially turn a database matrix around so that terms are regarded to have representations through their weighted associations to objects in the database. A similarity measure now works with terms of weighted objects rather than objects of weighted terms. Given, the symmetrical nature of a database matrix of object/term weights, it is reasonable to suppose that obser-
vations made in this article will apply with equal force in the computation of term (as opposed to object) similarity.

It is interesting to note that computations of term simi-

larity are automatically and dynamically factored into a spreading activation process that simply allows activation to spread a little further than it does in the spreading activation measure we have considered in this article. Using this article's spreading activation measure, activation spreads from the query to a set J of its associated terms in the first stage. It then spreads from these terms to a set K of associ-

ated database objects in a second stage.

Suppose we allow activation to spread from objects in set K to a set L of their associated terms in a third stage. Clearly, set L will contain the set J of original query terms (providing links are bi-directional). In addition, non-query terms are contained in this set that indirectly receive object-

mediated activation from the query. Terms receiving the most activation, in this manner, are terms that tend to co-
occur with query terms across database objects, i.e., these are terms with usages that are similar to those of the query terms. Therefore, we might reasonably include these new terms in the query in order to increase its recall rate. This is precisely what happens if activation is allowed to spread from terms in set L to their set M of object associates in a fourth stage of activation spread.
Stages two and three of this process can be regarded as a spreading activation computation of term synonymy or similarity which is applied to each term in the query. Stage four then makes this computation an integral part of the computation of object similarity. This process allows for a continuous variation in term similarity and it allows for the possibility that a term, in its various senses, may be related to a variety of other terms in the database. In these and other respects, the spreading activation approach compares favorably with a class of techniques involving the static construction of term synonym classes (see [20] for a more complete discussion of these comparisons).

5. Conclusion: The Geometric Approach

The utility of an information retrieval system critically depends upon the performance of its similarity measure. The performance of a similarity measure, in turn, will depend upon its semantic sensitivity to various kinds of information that may be contained in a database matrix of object/term weights. Important information may be contained, for example, in both within-object and between-object term weight relationships. Other useful information may be contained in patterns of term usage. However, the relationship between a similarity measure’s algebraic description (i.e., its computational formula) and its semantic sensitivity to database information is not always clear.

In this article we advance a geometric approach to the analysis of similarity measures that helps to bridge the gap between a similarity measure’s algebraic and semantic descriptions. Query and database objects can be represented in an n-space whose dimensions correspond to terms in the database. Using the geometric approach we attempt to characterize a similarity measure through the nature of its iso-similarity contours in this n-space. This analysis reveals striking differences among the similarity measures considered. We emphasize, again, that the practical significance of any such differences must be empirically validated; the geometric approach cannot take the place of such validation. However, the geometric approach may help to complement, and perhaps to guide, empirical analyses of similarity measures. Specifically, the geometric approach may help us to better determine those circumstances in which algebraic differences in similarity measure formulae will have an empirically verifiable impact upon retrieval performance.

Acknowledgments

We would like to thank Steve Bulick, Sandy Kelso, Tumi Landauer, Don Norman and Steve Poltrock for their helpful comments on earlier drafts of this article.

References