7.3.6 Similarity

Similarity is a measure of the resemblance between two things that share some characteristics but are not identical. It is a very flexible notion whose meaning depends on the domain within which we apply it. Some people consider that the concept of similarity is itself meaningless because there must always be some basis, some unstated set of properties, for determining whether two things are similar. If we could identify those properties and how they are used, there would not be any work for a similarity mechanism to do.[424][CogSci]

To make similarity a useful mechanism for categorization we have to specify how the similarity measure is determined. There are four psychologically-motivated approaches that propose different functions for computing similarity: feature- or property-based, geometry-based, transformational, and alignment- or analogy-based. The big contrast here is between models that represent items as sets of properties or discrete conceptual features, and those that assume that properties vary on a continuous metric space.[425][CogSci]

7.3.6.1 Feature-based Models of Similarity

An influential model of feature-based similarity calculation is Amos Tversky's contrast model, which matches the features or properties of two things and computes a similarity measure according to three sets of features:

- those features they share,
- those features that the first has that the second lacks, and
- those features that the second has that the first lacks.

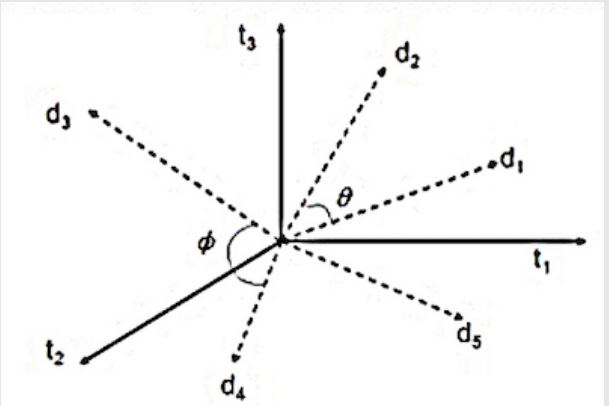
The similarity based on the shared features is reduced by the two sets of distinctive ones. The weights assigned to each set can be adjusted to explain judgments of category membership. Another commonly feature-based similarity measure is the Jaccard coefficient, the ratio of the common features to the total number of them. This simple calculation equals zero if there are no overlapping features and one if all features overlap. Jaccard's measure is often used to calculate document similarity by treating each word as a feature.[426][CogSci]

We often use a heuristic version of feature-based similarity calculation when we create multi-level or hierarchical category systems to ensure that the categories at each level are at the same level of abstraction or breadth. For example, if we were organizing a collection of musical instruments, it would not seem correct to have subcategories of "woodwind instruments," "violins," and "cellos" because the feature-based similarity among the categories is not the same for all pairwise comparisons among the categories; violins and cellos are simply too similar to each other to be separate categories given woodwinds as a category.

7.3.6.2 Geometric Models of Similarity

Geometric models are a type of similarity framework in which items whose property values are metric are represented as points in a multi-dimensional feature- or property-space. The property values are the coordinates, and similarity is calculated by measuring the distance between the items.

Document Similarity

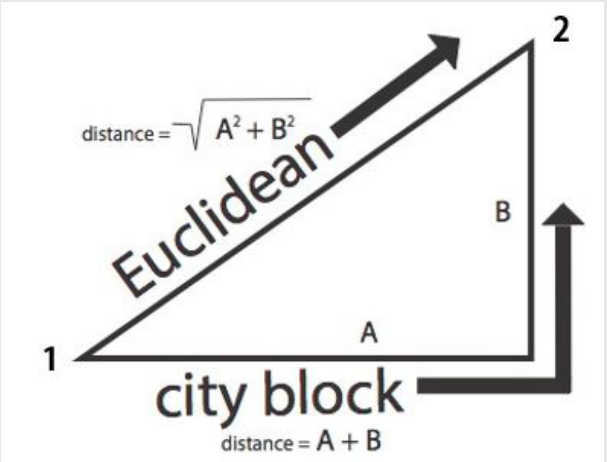


Documents represented as vectors in term space, with the angles between them as a measure of their similarity.

Geometric similarity functions are commonly used by search engines; if a query and document are each represented as a vector of search terms, relevance is determined by the distance between the vectors in the "term space." The simplified diagram in the sidebar, Document Similarity (page 378), depicts four documents whose locations in the term space are determined by how many of each of three terms they contain. The document vectors are normalized to length 1, which makes it possible to use the cosine of the angle between any two documents as a measure of their similarity. Documents d1 and d2 are more similar to each other than documents d3 and d4, because angle between the former pair (Θ) is smaller than the angle between the latter (Φ). We will discuss how this works in greater detail in Chapter 10, *Interactions with Resources*.

If the vectors that represent items in a multi-dimensional property space are of different lengths, instead of calculating similarity using cosines we need to calculate similarity in a way that more explicitly considers the differences on each dimension.

Geometric Distance Functions



The distance between points 1 and 2 depends on how the distance function combines the differences in values (A and B) on each dimension.

The diagram in the sidebar, Geometric Distance Functions (page 379) shows two different ways of calculating the distance between points 1 and 2 using the differences A and B. The Euclidean distance function takes the square root of the sum of the squared differences on each dimension; in two dimensions, this is the familiar Pythagorean Theorem to calculate the length of the hypotenuse of a right triangle, where the exponent applied to the differences is 2. In contrast, the City Block distance function, so-named because it is the natural way to measure distances in cities with "gridlike" street plans, simply adds up the differences on each dimension, which is equivalent to an exponent of 1.

We can interpret the exponent as a weighting function that determines the relative contribution of each property to the overall distance or similarity calculation. The choice of exponent depends on the type of properties that characterize a domain and how people make category judgments within it. The exponent of 1 in the City Block function ensures that each property contributes its full amount. As the exponent grows larger, it magnifies the impact of the properties on which differences are the largest.

The Chebyshev function takes this to the limit (where the exponent would be infinity) and defines the distance between two items as the difference of their values on the single property with the greatest difference. What this means in practice is that two items could have similar or even identical values on most properties, but if they differ much on just one property, they will be treated as very dissimilar. We can make an analogy to stereotyping or prejudice when a person is just like you in all ways except for the one property you view as negative, which then becomes the only one that matters to you.

At the other extreme, if the exponent is reduced to zero, this treats each property as binary, either present or absent, and the distance function becomes a count of the number of times that the value of the property for one item is different from the value for the other one. This is called the "Hamming distance."

7.3.6.3 Transformational Models of Similarity

Transformational models assume that the similarity between two things is inversely proportional to the complexity of the transformation required to turn one into the other. The simplest transformational model of similarity counts the number of properties that would need to change their values. More generally, one way to perform the *name matching* task of determining when two different strings denote the same person, object, or other named entity is to calculate the "edit distance" between them; the number of changes required to transform one into the other.

The simplest calculation just counts the number of insertion, deletion, and substitution operations and is called the Levenshtein distance; for example, the distance between "bob" and "book" is two: insert "o" and change the second "b" to "k". Two strings with a short edit distance might be variant spellings or misspellings of the same name, and transformational models that are sensitive to common typing errors like transposed or duplicated letters are very effective at spelling correction. Transformational models of similarity are also commonly used to detect plagiarism and duplicate web pages.[427][Com]

7.3.6.4 Alignment or Analogy Models of Similarity

None of the previous types of similarity models works very well when comparing things that have lots of internal or relational structure. In these cases, calculations based on matching features is insufficient; you need to compare features that align because they have the same role in structures or relationships. For example, a car with a green wheel and a truck with a green hood both share the feature green, but this matching feature does not increase their similarity much because the car's wheel does not align with the truck's hood. On the other hand, analogy lets us say that an atom is like the solar system. They have no common properties, but they share the relationship of having smaller objects revolving around a large one.

This kind of analogical comparison is especially important in problem solving. You might think that experts are good at solving problems in their domain of expertise because they have organized their knowledge and experience in ways that enable efficient search for and evaluation of possible solutions. For example, it is well known that chess masters search their memories of previous winning positions and the associated moves to decide what to play. However, top chess players also organize their knowledge and select moves on the basis of abstract similarities that cannot be explained in terms of specific positions of chess pieces. This idea that experts represent and solve problems at deeper levels than novices do by using more abstract principles or domain structure has been replicated in many areas. Novices tend to focus more on surface properties and rely more on literal similarity.[428][CogSci]