A Tutorial on Distance and Similarity

Abstract — This is an introductory tutorial on distance and similarity.
Keywords: distance, similarity, association measures
First published: 02-13-2015
© E. García; admin@minerazzi.com

Introduction

This tutorial was written as a companion for two of our tools (1, 2). These were developed to simplify distance and similarity calculations. In data mining and information retrieval (3 – 6), these are considered association measures where distance is lack of similarity and similarity is resemblance. Some authors prefer to use the term ‘dissimilarity’ instead of distance.

Distance

Distance is a metric. A function \( f \) is said to be a metric if it exhibits reflexivity, symmetry, and triangular inequality. Consider three points \( a, b, \) and \( c \) describing a triangle in a two-dimensional space.

- **Reflexivity** means that the distance from a point to itself is zero; e.g., \( f(a, a) = f(b, b) = f(c, c) = 0 \).
- **Symmetry** refers to the fact that the distance between any two points, measured from either one, is the same; e.g., \( f(a, b) = f(b, a) \).
- **Triangular inequality** means that the distance between any two points is equal or less than the distance between these measured through a third point; e.g., \( f(a, b) + f(b, c) \geq f(a, c) \).

If said conditions are not met, the function in question is not a metric. Distances cannot be negative and are not naturally upper bounded. Finally, we can arithmetically average, add, or subtract distances to compute new distances.

Similarity

Although they have symmetry, similarities are not metrics. They can be negative and upper/lower bounded. For example, Hamann, Yule, and Pearson’s Phi adopt values between -1 and +1. In addition, the similarity of a point or data set to itself is 1, \( s_{ii} = 1 \).

Similarity functions are used to measure the resemblance between points and data sets; i.e., how similar or alike they are. They cannot be arithmetically averaged, added, or subtracted to compute new similarities.

The best known similarity measure is the so-called cosine similarity. We may represent single points or entire data sets as vectors and compare the angles between them. As two vectors get closer to each other, the angle between them decreases and the cosine of said angle increases.

For unit vectors (vectors whose lengths have been normalized to equal 1), their dot product equals the cosine of the angle between them. For paired data consisting of z-scores (mean-centered paired data normalized by their standard deviations), their cosine similarity and Pearson’s correlation coefficient are the same thing. As cosines are not additive, this is a confirmation and reminder of the non-additivity of correlation coefficients.
Tools for Computing Distances and Similarities

As mentioned early in this tutorial, we have developed two tools for solving distance and similarity problems (1, 2). Both tools work by accepting any two data sets, provided that these are binary and of same size (same number of data points).

Consider the following data sets.

\[ A = \{1, 0, 1, 1, 0\} \]
\[ B = \{1, 1, 0, 1, 1\} \]

We may express their distance or similarity using one or several different frameworks, depending on the meaning of the 0's and 1's and the problem at hand. Our tools do this by generating a 2x2 contingency table consisting of the following \((i,j)\) counts:

- \((1,1)\) counts, meaning 'positive matches'.
- \((1,0)\) counts, meaning 'i absence mismatches'.
- \((0,1)\) counts, meaning 'j absence mismatches'.
- \((0,0)\) counts, meaning 'negative matches'.

Those familiar with 2x2 contingency tables know that for binary data the diagonal from \((1,1)\) to \((0,0)\) represents the total number of matches or "correct answers" between \(i\) and \(j\).

By contrast, the diagonal from \((0,1)\) to \((1,0)\) represents the total number of mismatches or "incorrect answers" between \(i\) and \(j\). This is the so-called Hamming Distance.

The Hamming Distance is a measure that only allows substitutions and applies to sets of same size. For binary sets of same size, the Hamming, Manhattan (City-Block), and Squared Euclidean (\(D^2\)) distances are all the same thing. For same size sets, Hamming Distance is an upper bound on the Levenshtein Distance.

Distance-Similarity Transformations

As noted by Lin (3), the definition of similarity depends on the model or knowledge domain under inspection and is tied to a specific problem or application. Arbitrarily transforming similarities into distances and vice versa simply compounds many of the problems described by Lin and others (3, 5) and can induce to error.

Sometimes such transformations are done using the following trick of the trade.

\[ S = 1 - D \quad \text{(Eq 1)} \]

This is how Jaccard, Dice, Sokal-Michener, Rogers-Tanimoto, and Rusell-Rao similarities are frequently transformed into distances (4). Eq 1 suggests a complementary relationship between \(S\) and \(D\). However, this is not always the case. Actually, some authors use (3, 5)

\[ S = 1/(1 + D) \quad \text{(Eq 2)} \]

or
\[ S = 1 - D^2 \]  
(Eq 3)

In general, similarity-distance transformations are determined by the problem to be solved.

As stressed by Toit, Steyn, and Stumpf (5), while a distance can be transformed into a similarity with Eq 2, the reverse process is not so obvious because of the triangular inequality which must be satisfied by a distance metric.

So given a similarity matrix \( S \) populated with \( S_{ij} \) values: How could we compute the corresponding distance matrix \( D \)? Assuming that the similarity matrix \( S \) is positive semi-definite

\[ D_{ij} = \sqrt{S_{ii} - 2S_{ij} + S_{jj}} \]  
(Eq 4)

is the standard transformation from \( S \) to \( D \). For the particular case of \( S_{ii} = S_{jj} = 1 \),

\[ D_{ij} = \sqrt{2 - 2S_{ij}} \]  
(Eq 5)

Eq 5 can be used to compute \( D_{ij} \) and matrix \( D \).

**Conclusion**

Distances and similarities have symmetry. However, distances are always positive while similarities can be negative. Distances are metrics, while similarities are not. We can arithmetically average, add, or subtract distances to compute new distances, but we cannot do the same with similarities. If similarity is resemblance and distance is lack of resemblance, then “similarity distance” (6) is an oxymoron.

**References**