**Project 3: Dissimilarities between data objects**

Suraj Eswaran

University Of The Cumberlands

2024 Spring – Statistics For Data Science (MSDS – 531 – A01) – First Bi-Term

Professor Dr. Danny T. Barnes

02/18/2024

**This project demonstrates how to measure similarities between data objects. These topics described are mostly in chapter 6 Statistical Machine Learning from ‘Practical Statistics for Data Scientists’. Cover in the project the following:**

**Find some data examples and show examples of calculating**

**Euclidean distance**

Euclidean distance is calculated with the square root of the sum of the squared differences between corresponding coordinates of the two points across all dimensions (Suwanda et al., 2020). First, we will consider 2-Dimensional example in a 2-Dimensional space where point A is coordinated at (3,4) and point B is coordinated at (6,8), thus Euclidean distance between these two points calculated by sum of square root of square of (6-3) which is equal to 9 and square of (8-4) which is equal to 16, and sum of those two value is equal to 25 and square root of 25 is 5. Thus, the value of Euclidean distance is equal to 5.

Let's consider a real-world example where we calculate the Euclidean distance between two cars based on their fuel efficiency and price. So, Car A having Fuel efficiency 30 mpg with price of $25,000 and Car B having Fuel efficiency of 35 mpg with price of $35,000. Euclidean distance between these two cars based on these features is calculated by sum of square root of square of (30-35) which is equal to 25 and square of (25000-35000) which is equal to 100,000,000, and sum of those two value is equal to 100,000,025 and square root of 100,000,025 is approximately 10,000.50. Thus, the value of Euclidean distance based on fuel efficiency and price is equal to 10,000.50.

**L1 distance**

L1 distance is a measure of the distance between two points in a grid-based space. It is calculated by adding the absolute differences of the coordinates of the two points (Stone, 1991). Let’s consider a simple example where point A is coordinated at (2,3) and point B is coordinated at (5,8), thus L1 distance is calculated by finding the absolute difference of (2-5), which is equal to 3 and absolute difference of (3-8), which is equal to 5 and total of them will result in 8. Thus, the L1 distance between point A and point B is equal to 8.

Another example where we calculate the L1 distance between two cars based on their location on a grid where Car A is located at (3,8) and Car B is located at (7,2). L1 distance is calculated by finding the absolute difference of (3-7), which is equal to 4 and absolute difference of (8-2), which is equal to 6 and total of them will result in 9. Thus, the L1 distance between Car A and Car B is equal to 9.

**Prove or disprove that Euclidean and L1 distance satisfy**

**Positivity**

**d(x,y) >= 0 for all x and y,**

**d(x,y) == 0 only if x == y.**

First, we will try to validate Euclidean Distance for condition “**d(x,y)>=0 for all x and y**” and “**d(x,y) == 0 only if x==y**”. Euclidean Distance is calculated by square root of the sum of the squared differences between corresponding coordinates of the two points across all dimensions (Suwanda et al., 2020). Since we are taking the square root of the sum of squared differences, each term in the sum is non-negative. Therefore, the sum is also non-negative. Always the square root of a non-negative number will result in a non-negative answer. Thus, it will satisfy the condition “**d(x,y)>=0 for all x and y**” successfully. For condition “**d(x,y) == 0 only if x==y**”, where the value of x is equal to y, so sum of squared values will be equal to zero and thus Euclidian Distance will be equal to zero. So, “**d(x,y)(Euclidean Distance)==0”** will be satisfied only when if all corresponding coordinates of x and y are equal.

Now, we will try to validate L1 Distance for condition “**d(x,y)>=0 for all x and y**” and “**d(x,y) == 0 only if x==y**”. L1 Distance is calculated by sum of absolute difference of the coordinates of the two points (Stone, 1991). Each term |x-y| is non-negative since it results the absolute difference between x and y, thus summing up non-negative terms results in a non-negative value. So, L1 Distance satisfy the condition “**d(x,y)>=0 for all x and y**”. The L1 distance “**d(x,y) == 0 only if x==y**” will be equal to zero if and only if x value is equal to y value. This is because if

x is equal to y, then each term in the sum |x-y| will be zero, resulting in a sum of zeros. The sum of zeros is zero. So, L1 Distance satisfy the condition “**d(x,y) == 0 only if x==y**”.

**Symmetry**

**d(x,y) == d(y,x) for all x and y.**

Now, we will try to validate Euclidean Distance and L1 Distance for condition “**d(x,y) == d(y,x) for all x and y.**”. As Euclidean Distance is the square root of the sum of squared differences between coordinates of two points (Suwanda et al., 2020), so **“d(x,y)(Euclidean Distance)”** is calculated by square root of square of difference (x-y) and **“d(y,x)(Euclidean Distance)”** is also calculated by square root of square of difference (y-x). Since the square of a real number is always non-negative, and the square root function is also non-negative, we can conclude that the condition “**d(x,y) == d(y,x) for all x and y.**”.

L1 Distance is calculated by sum of absolute difference of the coordinates of the two points (Stone, 1991), so **“d(x,y)(L1 Distance)”** is calculated by sum of absolute difference (x-y) and **“d(y,x)(L1 Distance)”** is also calculated by sum of absolute difference (y-x). We can see that **“d(y,x)”** is indeed the same as **“d(x,y)”** in this case, as the order of subtraction inside the absolute values does not change the result. Hence, L1 distance satisfies the symmetry property**.**

**Triangle Inequality**

**d(x,z) <= d(x,y) + d(y,z) for all points x, y, and z**

Euclidean Distance of **d(x,z)** is equal to square root of square of (x-z), **d(x,y)** is equal to square root of square of (x-y) and **d(y,z)** is equal to square root of square of (y-z) (Suwanda et al., 2020). Now, lets square both the side of the equation for eliminating the square root, results in square of (x-z) <= square of (x-y) + square of (y-z). We will try expanding the equation by using mathematical formula, square of (a+b) equals to square value of ‘a’ plus square value of ‘b’ plus product of ‘a’, ‘b’ and 2, it will be showing square value of ‘x’ plus square value of ‘z’ plus product of ‘x’, ‘z’ and 2 <= square value of ‘(x-y)’ plus square value of ‘(y-z)’ plus product of ‘(x-y)’, ‘(y-z)’ and 2. Thus, simplifying and cancelling out terms, we were getting 0<= product of 2 with summative of product of (x-y) with (y-z). From the result, we could understand that this inequality properly will be true because squares are always non-negative, so the sum of squares is non-negative, and hence the inequality **d(x,z) <= d(x,y) + d(y,z) for all points x, y, and z** will be true, thus it satisfies in Euclidian Distance.

In the case of L1 Distance, where we will calculate absolute difference two coordinates, so **d(x,z)** will be equal to |x-z|, **d(x,y)** will be equal to |x-y| and **d(y,z)** will be equal to |y-z|. Replacing on to the equation, |x-z| <= |x-y| plus |y-z|. If we examine this inequality, we can see that each term in the sum on the left side is less than or equal to the corresponding terms in the sums on the right side due to the properties of absolute values. Therefore, we can result that the L1 distance satisfies the triangle inequality.

**Explain why it is not possible or why it is possible to**

**rearrange data so Euclidean distance gives the same meaning as Hamming distance.**

Euclidean distance basically measures the distance between points in a Euclidean space (Suwanda et al., 2020). It is sensitive to the scale and relative positioning of data points. But Hamming distance measures the number of positions differ between two characters of equal length, without considering the magnitude of its differences (Kolpakov & Kucherov, 2003). It simply counts the number of mismatches between categorical or binary data points. Even if one could technically rearrange the data to make Euclidean distance equal to Hamming distance, the resulting distances would likely have different interpretations and applications. Hence, it's unlikely that rearranging the data would allow Euclidean distance to give the same meaning as Hamming distance. Thus, it is not possible to rearrange data so Euclidean distance gives the same meaning as Hamming distance.

**show that measure d=1-cos(x,y) satisfies positivity, symmetry, and triangle Inequality**

First, we will try to validate whether d=1-cos(x,y) satisfies positivity where the measure should show non-negative result. At any angle, the range of cos function will be between 1 to -1, so 1-cos(x,y) will also be between 1 to -1. For example, consider x to be equal to 2 and y to be equal to 5, cos function formula is equal to product of x and y divided by absolute value of x multiply by absolute value of y. So, calculating those value, we have found out that cos value is equal to 1 and d is equal to 1 minus 1, which is equal to zero. Thus, we can infer from the observation that the measure **“d=1-cos(x,y)”** satisfies positivity property.

Now, let’s take the same example to prove symmetry property where x is equal to 2 and y to be equal to 5. So, cos function formula is equal to product of x and y divided by absolute value of x multiply by absolute value of y, so cos(2,5) and cos(5,2) will be equal and thus, “**d=1-cos(x,y)”** for value cos(2,5) and cos(5,2) will be equal. Thus, the distance between two points

x and y are the same as the distance between y and x in one-dimensional space, so it does follow symmetry rule.

In order to prove Triangle Inequality for **“d=1-cos(x,y)”**, where we are using x equal to [1 0], y equal to [0 1] and z equal to [1 1]. For “**d(x,y)**”, we will get cos(x,y) with the help of cosine formula will be equal to zero, thus will result in 1. For “**d(y,z)**”, we will get cos(z,y) with the help of cosine formula will be equal to 1/ √2 and, thus will result in 1 - 1/ √2. For “**d(x,z)**”, we will get cos(x,z) with the help of cosine formula will be equal to 1/ √2 and, thus will result in 1 - 1/ √2. Now, validating Triangle Inequality **d(x,z) <= d(x,y) + d(y,z)**, so 1- 1/ √2 <= 1+ (1/ √2), where we will finally get 1/ √2 <= 2/ √2, thus it satisfies the condition.

**Draw conclusions about what is important when choosing the distance measure for the evaluation of dissimilarities between data objects.**

In order to select the distance measure for evaluation of dissimilarities between data objects, we should make sure that measure should satisfy metric properties such as positivity, symmetry, and the triangle inequality. These properties ensures that the distance function are functioning consistently so that they follow some kind of a unity. Also, the choice of measure depends on the characteristics and requirement of the context where different application will require different distance measures. In image processing applications such as image retrieval or image clustering, distance measures like Euclidean distance or L1 distance are commonly used where it can quantify the similarity between image features such as color histograms or texture descriptors. Thus, the choice of distance measure should be guided by the specific characteristics of the data, the underlying problem, and the desired properties of the distance metric, such as interpretability, robustness, and computational efficiency.

**References**

Suwanda, R., Syahputra, Z., & Zamzami, E. M. (2020, June). Analysis of euclidean distance and manhattan distance in the K-means algorithm for variations number of centroid K. In Journal of Physics: Conference Series (Vol. 1566, No. 1, p. 012058). IOP Publishing.

Stone, R. E. (1991). Some average distance results. Transportation Science, 25(1), 83-90.

Kolpakov, R., & Kucherov, G. (2003). Finding approximate repetitions under Hamming distance. Theoretical Computer Science, 303(1), 135-156.