On Integral Metrics and Trajectory Classification

Lillian Vernor

Abstract

In this project, we explore distance in the context of metrics. More specifically, we take a look at an integral metric that is used to determine the distance between sets. The motivation behind this project is to determine that the integral metric we use is meaningful in the context of our data set. The data set we use consists of trajectories of cars along a portion of the I5 highway. Through training, testing and evaluating this model on the data set, we can reach conclusions on the structure of the data and the success of this integral metric in terms of a classifier. In the end, we can both determine whether or not this integral metric fits with the data set chosen and explore other areas where the metric could possibly succeed or where it might fail.

Chapter 1

Introduction

In the data analysis, the task of classification involves identifying the category that new data will fall under. This can provide a look into the structure of the data and any patterns it may have. We consider the use of integral metrics as a classifier. In a simple sense, a metric is used to determine distance. For our purposes, we look at an integral metric to capture the distance between two sets. Through this evaluation, we are able to explore what it is that this integral metric truly does and how it looks at distance. By comparing it to other metrics, we can see what makes this integral metric unique. Additionally, we can see in what context this integral metric is most successful and where it might fail. Our end goal is to determine whether or not this metric can be useful on the chosen data set.

The data set on which we evaluate the performance of this integral metric includes a series of trajectories (pathways of the movement of an object) of cars on a portion of the I5 highway. Each trajectory represents the movement of a car in a lane on the highway. For our purposes, we use this new way to define distance as a model for how we can break up this data. We are able to see how this integral metric can decipher the different classes within the data. This "breaking up" of the data leads to our task of the classification of trajectories. We evaluate how well our integral metric is able to determine which class each of the trajectories belongs to, based on the given labels of data. Using distance matrix computations and clustering methods, we can see the level of success that this integral metric has in terms of trajectory classification. After using a validation method, we can come to a conclusion on how this metric works with the data set.

Chapter 2

Mathematical Background

There are many different ways to describe distance. For example, think of the distance between your home and a convenience store. One way to measure this distance is what is commonly known as the taxicab distance, which is the distance that is taken by a car traveling square blocks. Another common way to measure distance is what is often thought of as the shortest distance, sometimes called the "usual distance" or straight-line distance. In mathematics, a generalization of distance can be captured by what is called a metric.

DEFINITION 2.0.1. *A* metric d *on a set* X *is a function* $d : X \times X \to \mathbb{R}$ *such that for all* $x, y \in X$*:*

(1) $d(x, y) \geq 0$

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

(3) $d(x, y) = d(y, x)$ *(symmetry)*

(4) $d(x, y) \leq d(x, z) + d(z, y)$ *(triangle inequality).*

A metric space (X, d) is a set X with a metric d defined on X where d has all of the properties as above.

This first example is something you might have seen in elementary school. This metric describes the distance between two numbers on the number-line.

Example 2.0.1. This example calculates the distance between two quantities in $\mathbb{R}$, and the absolute value of this difference is taken because distance is always positive.

$$d(x, y) = |x - y|$$

One of the most common metrics is the Euclidean metric, used to calculate the distance between points. This metric is referred to as the usual distance, or straight-line distance, mentioned previously. The Euclidean metric differs from the above metric because it represents distance in $\mathbb{R}^2$.

Example 2.0.2. Euclidean Distance. This is the formula for Euclidean distance in $\mathbb{R}^2$.

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The following example, known as the taxicab metric, represents the distance a car might travel around blocks of a city.

Example 2.0.3. The taxicab metric is the metric of $\mathbb{R}^2$ defined by:

$$d((x_1, y_1), (x_2, y_2)) = |x_1 - x_2| + |y_1 - y_2|$$

for all points. It is equal to the total length of any path connecting these two points along vertical and horizontal line segments [4].

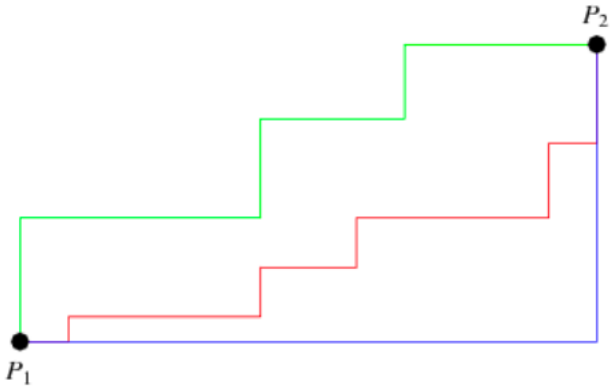


Figure 2.1. Taxicab Metric

However, what if we would like to think of distance in a different way? What if we desire a distance between sets that are more than just single points? For example, the distance between two lines or two curves. The Hausdorff distance, Example 2.0.4, is one way to calculate the distance between such sets. Later, we examine the integral metric as another way to measure the distance between sets.

Example 2.0.4. The Hausdorff Distance. This metric is used to calculate the distance between two sets. Informally, two sets are close in the Hausdorff distance if every point of either set is close to some point of the other set. The Hausdorff distance is the longest distance you can be forced to travel, from a point in one set to another point in the other set. In other words, it is the greatest of all the distances from a point in one set to the closest point in the other set. Let X and Y be two non-empty subsets of a metric space (M, d). We define their Hausdorff distance $d_H(X, Y)$ by:

$$d_H(X,Y) = \max\{\sup_{x\in X}\inf_{y\in Y} d(x,y), \sup_{y\in Y}\inf_{x\in X} d(x,y)\}$$

Figure 2.2 demonstrates two rings which represent two sets [8].

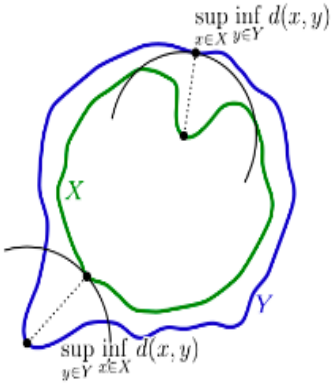


Figure 2.2. Hausdorff Distance [8]

2.1. Integral Metrics

In this paper, we focus on a type of metric called an integral metric. Much like the Hausdorff distance, it is used to calculate the distance between two sets. Charatonik and Insall (see [10]) define the following:

DEFINITION 2.1.1. *Let (X, d) be a compact metric space, $A \subset X$, $B \subset X$, and define $d_A : X \to [0, \infty)$ by*

$$d_A(x) = d(x, A) = \inf\{d(x, y) | y \in A\}.$$

Then for a finite measure λ and $p \in (0, \infty)$,

$$D_p(A, B) = \left(\int_X |d_A(x) - d_B(x)|^p d\lambda(x) \right)^{\frac{1}{p}}.$$

If $p \geq 1$ and λ is strictly positive then $D_p(A, B)$ is a metric (as proved in [10], Theorem 3.2). We will only consider the Lebesgue measure, which meets this criteria. The following example shows a numerical application of this integral metric.

EXAMPLE 2.1.1. *Let $X = [0, 1]$, $A = \{0\}$, $B = \{1\}$, and let d denote the usual metric on the real unit interval: $d(x, y) = |y - x|$ and let $p = 1$.*

$$\begin{aligned} D_p(A, B) &= \int_0^1 ||0 - x| - |1 - x|| d\lambda(x) \\ &= \int_0^1 || - x| - |1 - x|| d\lambda(x) \\ &= \int_0^1 ||x| - |1 - x|| d\lambda(x) \\ &= \int_0^1 |2x - 1| d\lambda(x) \end{aligned}$$

Now we split up the integral into the sum of two integrals.

$$\begin{aligned} D_p(A,B) &= \int_0^{1/2} 1-2x d\lambda(x) + \int_{1/2}^{1} 2x-1 d\lambda(x) \\ &= x-x^2\Big|_0^{1/2} + x^2-x\Big|_{1/2}^{1} \\ &= 1/4 + 1/4 \\ &= 1/2 \end{aligned}$$

For $p=\infty$ and λ strictly positive (according to [10] Observation 3.2) $D_p(A,B)$ is

$$D_\infty(A,B) = max\{|d_A(x) - d_B(x)||x \in X\}$$

which is just simply the Hausdorff distance. One of the main differences between the integral metric and the Hausdorff metric is that the integral metric accounts for the underlying compact metric space X. Informally, the integral over X can be thought of as the infinite sum of the differences between the distance from each point in X to the sets A and B. In fact, if we approximate X with a finite set, then the integral can be approximated with a Riemann sum.

More specifically, we will only consider the special case where $A, B \subseteq X \subset \mathbb{R}^2$, $d(x,y)$ is the standard Euclidean distance, and λ is the Lebesgue measure. If $X = (a,b) \times (c,d)$ then

$$D_p(A,B) = \int_a^b \int_c^d |d_A(x) - d_B(x)|^p dxdy$$

which follows from Fubini's Theorem (this theorem can be found in any Calculus 3 textbook). To approximate $D_p(A, B)$ with a Riemann sum, we partition the rectangular space X into a set of n × n squares. For each of the "center" points of the squares, the integrand's value at the center is multiplied by the area of the n × n box to represent the volume; then the sum of these volumes is an approximation of $D_p(A, B)$.

2.2. Clustering

Cluster analysis is one way to explore data by grouping a set of objects so that objects in the same group are more similar to each other than those of the other group. There are many different clustering algorithms used for data analysis. The purpose of cluster analysis can be to discover the success with which a particular clustering model can determine a difference between members of a set. In the context of supervised learning, clustering can be used to classify objects to given labels, whereas in unsupervised learning, the purpose is merely to learn more about the structure of the data without having specific labels [9].

K-medoid clustering is a method that uses the partitioning around medoids (PAM) algorithm. In general, the PAM algorithm works by first choosing k data points to be what are called the medoids, and then associating each data point to which ever medoid it is closet too. Then, for each medoid, it considers the swap of the medoid with another data point in the group and computes the cost change. If the cost decreases, the swap is performed. If not, the algorithm is complete and the best solution is found for that particular set of initial medoids [1].

2.3. Model Validation

Model validation is the process of ensuring that the model serves its intended purpose. In the case of classification tasks, including trajectory classification, the goal is to use the model to accurately determine the proper class to which each element in the data set belongs.

K-fold cross validation is a validation method with the purpose to test how accurately a model will perform on an independent data set. The first step is to randomly shuffle the data set and then break up the data into k sets. For each of the k-groups, set one group aside as a testing group and then fit the chosen model to the rest of the separated data as a training group. After the given model is fit to the training group, evaluate the model on the group that was set aside as a testing group and record its evaluation score. This is then repeated for each k-set. Lastly, compare all of the evaluation scores for each run and summarize conclusions [2].

Chapter 3

Using the Integral Metric on Data

3.1. Small-Scale Examples

Example 3.1.1. To illustrate a general way in which this integral metric works, the following example is of four horizontal lines which are equally spaced from each other. The purpose of this example is to show exactly what this metric is expected to decipher in terms of distance.

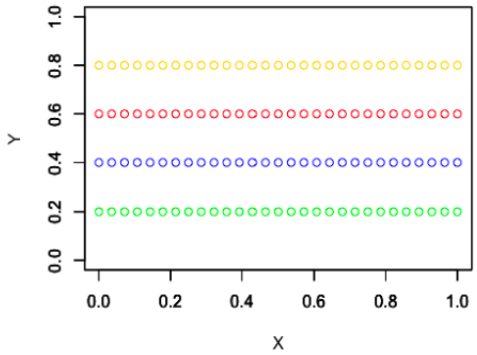


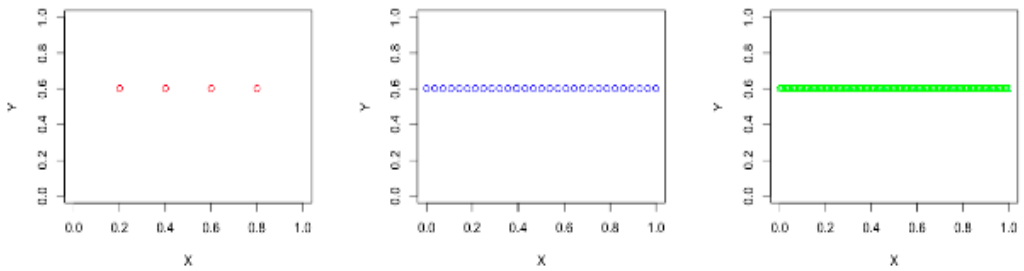
Figure 3.1. Stimulated Parallel Lines

Before running the distance matrix, we would expect based on Figure 3.1 that the two farthest lines would be the yellow and green.

	Green	*Blue*	*Red*	*Yellow*
Green	*0*	*0*	*0*	*0*
Blue	*.2036166*	*0*	*0*	*0*
Red	*.378082*	*.2035832*	*0*	*0*
Yellow	*.5228900*	*.3797825*	*.2035926*	*0*

As shown in the distance matrix, the greatest distance is between the green and yellow lines. The distance matrix also shows that any two lines next to each other are equally as close, due to how they are equally spaced on the graph. Because of this, we see symmetry in the matrix. In this example, we used n × n boxes of side length 0.05 and p value of 2.

Example 3.1.2. This example displays another insight we have gathered about this metric. This shows that the number of points or the size of the sample can affect the accuracy of this metric. We ran three plots, two densely populated and one sparse example, and ran a distance matrix over these three.



(A) (B) (C)

Figure 3.2. Simulated Representation of Variation in Sample Size

	(A)	*(B)*	*(C)*
(A)	*0*	*0*	*0*
(B)	*.04793681*	*0*	*0*
(C)	*.04763273*	*.00006381385*	*0*

The distance matrix shows that there is a difference in distance between these three examples. All of these data sets lie on the same line, but only differ in the number of points in the set. As you can see from the distance matrix, sets (B) and (C) are closest with only a small distance of .00006. Set (A) has size = 4. Set (B) has size = 29 and set (C) has size = 101. We used $n \times n$ boxes of side length 0.05 and p value of 2. From this example, we have learned that this metric has the ability to decipher sets of different size, even if they have the same shape.

3.2. Trajectory Classification

A trajectory can be described as the path representing the motion of an object in a certain space. There are many different ways to represent a trajectory depending on the information that it is intended to capture. A trajectory can be a vector that also has a velocity and time component. Or, a trajectory can be used to simply represent the position of an object at a moment in time. More specifically, a trajectory is a curve that can be approximated by a set of points that is used to represent the motion of an object. For our purposes, let a trajectory be the following:

$$T_i = \{(x_1, y_1), \dots (x_i, y_i)\}$$

Where the x, y coordinates represent position in $\mathbb{R}^2$ at different points in time.

$$T_i \subset X_i = (\min x_i, \max x_i) \times (\min y_i, \max y_i) \subset \mathbb{R}^2$$

Trajectory classification is the process of identifying trajectories of similar motion. Although the specific data set or classification method may differ, the goal remains to find similarities amongst the motion of objects. See [7] for more information on the different types or "levels" of trajectories and what they represent in terms of classification.

The data we use in this project comes from [6] and contains six different scenes. There are three simulated highway scenes, one actual highway scene, one actual intersection scene and one scene from the inside of a lab. The data we focus on is the simulated highway scene. The actual highway scene is from a portion of the I5 highway outside of UCSD. The three simulated scenes are supposed to mimic this actual data. For each of the 3 out of the 4 highway scenes, there are 8 classes representing the 8 lanes of traffic. This data set has two components: the 'tracks' which represent each individual trajectory and the 'truths' which contain the designated class label (1-8) for each trajectory. We utilize this labeled data to see if our model will correctly classify each trajectory based on the truth labels given in the data.

3.3. Simulated Interstate Data

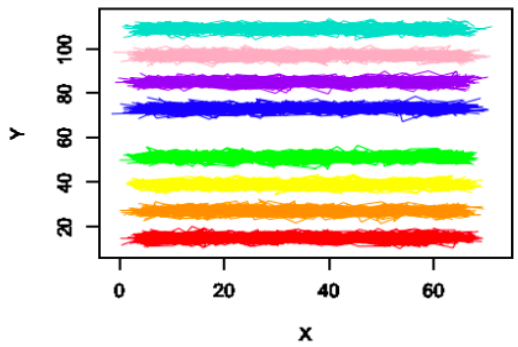


Figure 3.3. Plotted I5 Simulated Data

Let's consider the I5 simulated data (I5Sim), which approximates the trajectories of cars on the I5 freeway.

We use the k-fold cross validation method, with k = 4 on a subset of the data set of size 200. Each of our training sets of data are of size 150, while our testing data is of size 50. On each of the training sets, we use the integral metric to define distances between trajectories. With parameters p = 2 and n × n boxes of side length 1, we use a k-medoid clustering method PAM to build a model with eight clusters [3] (PAM is implemented in the R package [5]). Using the medoids chosen from the PAM algorithm on the training set, we compare the distance from each of the trajectories in the testing set and associate each trajectory with its closest medoid.

Lastly, in order to evaluate the proficiency to which this method classifies each trajectory, we create a confusion matrix where we are able to see the accuracy of classification. We identify the truth label of each of the trajectories and compare this truth label to the cluster that our training data assigned this trajectory. For this particular data, we see a perfect classification due to the simulated nature of the data. We can see in the sum of the four confusion matrices that all of the data lies on the diagonal, meaning that the predicted cluster from the k-medoids method matches the truth cluster. Our model clusters each of the 200 trajectories into the correct cluster based on the truth labels provided with the data.

$$\begin{bmatrix} 22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 22 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 22 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 32 \end{bmatrix}$$

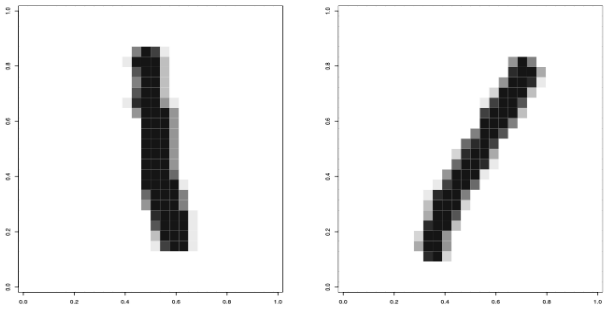
Chapter 4

Conclusions and Future Work

Using the data set of trajectories as well as some simple examples, we were able to gain insight into how this metric works and where it is most successful. Our investigation suggests that integral metrics could be a useful tool for trajectory classification. When using the I5 simulated data, the metric classified all trajectories with 100 percent accuracy. Additionally, in the more simple examples, the metric is shown to distinguish between horizontal lines in different places in the plane, as well as distinguish between the same line but with different sample sizes.

While the integral metrics ability to distinguish between lines in different places in the plane is important for trajectory classification, this can make other classification tasks more challenging. For example, we attempted to use the integral metric to classify the MNIST data set of handwritten digits [11]. The goal was to get 10 total classes for the digits 0-9. However, we noticed that this metric separated some of the digits into two classes. For example, the ones became two separate clusters. One of the clusters had ones that were slanted to the right and the other cluster of ones were slanted to the left, Figure 4.1 is an example. For the purpose of this classification, this was not the intention. It has been suggested by Matt Insall that this problem could be fixed by simply adjusting the p value in the metric, we hope to explore this in the future.

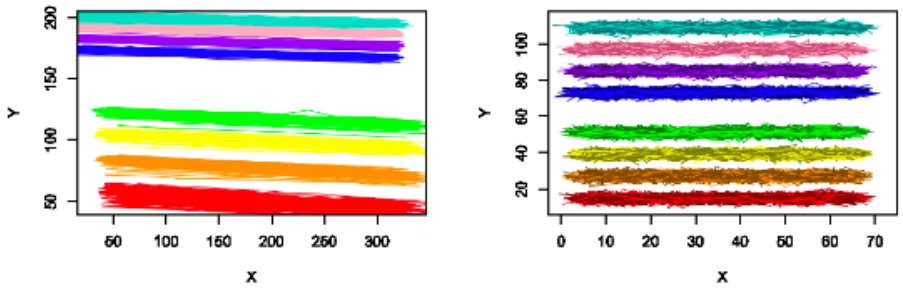
We also hope to further explore using the integral metric on the trajectory data set highlighted in this paper. Our intention is to run the same process we used in this paper on the real I5 data set that is not as clean as the simulated data set (see Figure 4.2) and determine how successful the metric would be in classifying. Additionally, the I5 sim 3 data



(A) (B)

Figure 4.1. Handwritten Ones from MNIST Data Set [11]

set has 16 classes where each lane of traffic has a class for slow cars and a class for fast cars. The difference is seen in the sample size of the trajectories where the slower cars have fewer data points and the faster cars have more data points. We predict this metric to be successful in this task because of the conclusions drawn from the basic example, where the metric is able to distinguish between different sample sizes of the same line.



(A) I5 Data (B) I5sim3 Data

Figure 4.2. Additional 15 Data

Bibliography

Betanzos, A., and Canedo, B. "Machine Learning; Studies from University of A Coruna Describe New Findings in Machine Learning (Big-Data Analysis, Cluster Analysis, and Machine-Learning Approaches)." Journal of Robotics & Machine Learning, 2019, pp. 541. https://doi.org/10.1007/978-3-319-77932-4_37

Jung, Y., and JH Hu. "A K-Fold Averaging Cross-Validation Procedure." Journal of Nonparametric Statistics, vol. 27, no. 2, 2015, pp.167-179. https://doi.org/10.1080/10485252.2015.1010532

Kaufman, Leonard. Finding Groups in Data: An Introduction to Cluster Analysis, United States, 1990.

Krause, E. F. Taxicab Geometry: An Adventure in Non-Euclidean Geometry. New York: Dover, 1986. https://doi.org/10.2307/3618288

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K. Cluster: Cluster Analysis Basics and Extensions. package version 2.1.0., 2019.

Morris, B., and Trivedi, M. Learning Trajectory Patterns by Clustering: Experimental Studies and Comparative Evaluation, 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 312-319, 2009. https://doi.org/10.1109/CVPR.2009.5206559

Morris, B., and Trivedi, M. Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and Long-Term Adaptive Approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(11), 2011. https://dio.org/10.1109/TPAMI.2011.64

Rocchini, "Example of Hausdorff distance", https://commons.wikimedia.org/wiki/ File:Hausdorff_distance_sample.svg, 2007.

Soni, Devin. Supervised vs. Unsupervised Learning Towards Data Science, 22 Mar. 2018. https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d

W. J. Charatonik and M. Insall Metrics on Hyperspaces: A Practical Approach Topology Proceedings Vol. 30, No. 1, pgs. 129-152, (2006). http://topology.auburn.edu/tp/reprints/v30/tp30109.pdf

Y.LeCun, L. Bottou, Y. Benigo, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, November 1998. https://doi.org/10.1109/5.726791