Persistent homology classification algorithm

Mark Lexter D. De Lara Corresp. 1, 2

1 Institute of Mathematical Sciences and Physics, College of Arts and Sciences, University of the Philippines Los Baños, College, Los Baños, Laguna, Philippines

2 Institute of Mathematics, University of the Philippines Diliman, Quezon City, Metro Manila, Philippines

Corresponding Author: Mark Lexter D. De Lara
Email address: mddelara@up.edu.ph

ABSTRACT

Data classification is an important task in machine learning, used to solve problems in numerous settings. There are many classifiers, but none of the algorithms work best for all kinds of data, as implied by the no free lunch theorem. Topological data analysis is a rapidly growing field that deals with the shape of data. One primary tool in this field used to analyze the shape or topological properties of a dataset is persistent homology, a method based on algebraic topology for computing topological features of a space of points which persists across multiple resolutions.

This study proposes a supervised learning and classification algorithm using persistent homology of training data classes in the form of persistence barcodes and diagrams to predict the output category of new observations. The developed algorithm was validated using real-world datasets and a synthetic dataset. The performance of the proposed classification algorithm on these datasets was compared to that of the most commonly used classifiers. Validation runs showed that the proposed persistent homology classification algorithm performed at par if not better than most of the classifiers considered.

INTRODUCTION

Machine learning is a major branch of artificial intelligence. It deals with the study of computer systems and computer algorithms that can automatically learn and improve from experience without being explicitly programmed to do so. It focuses on the development of computer programs that can process data and give predictive analysis. Machine learning techniques are generally divided into three major categories, namely supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, a system learns from a readily available training set of data with correctly labeled observations. One of the major tasks or problems addressed by supervised learning is classification.

Deleted: which
Deleted: has
Deleted: the ability to

Classification is the process of identifying, recognizing, grouping, and understanding new objects into categories/sub-populations (Alpaydin, 2014). A training dataset is composed of individual observations or n-dimensional data points which are split into an (n-1)-dimensional input vector often called features/explanatory variables, and into one-dimensional output vector/class/ label. These observations, also called instances, can be univariate, bivariate, or multivariate. These features also called attributes, are quantifiable properties that can be categorical, ordinal, integer-valued, or real-valued. A classification algorithm, also called a classifier, is a procedure that implements classification tasks. Moreover, the term classifiers may also refer to the mathematical function that maps input features to an output category.

Deleted: ,
Deleted: which

Classification algorithms have found many applications in the fields of computer vision, speech recognition, biometric identification, biological classification, pattern recognition, document classification, credit scoring, and many more. For instance, in medicine, the task of assigning a diagnosis to a given patient based on gathered features like age, gender, body mass index, presence of particular symptoms, etc., is a classification application. Classification problems can be categorized into binary classification or multi-class classification problems. Binary classification is the task of assigning an observation to exactly one of two categories, while multi-class classification is the process of assigning an instance to exactly one class out of more than two classes. Classification tasks tend to be harder in the presence of more than classes or more attributes.

The study of classification algorithms is a vast field. Since the rise of artificial intelligence, numerous classification algorithms have been developed. Several of these techniques can be used to solve binary classification problems. Some algorithms are specially developed to solve binary classification problems, while there are algorithms that can be used to solve binary and multi-class classification problems. Many of these multi-class classifiers are extensions or modifications of one or more binary classifiers.

The no free lunch theorems proved by David Wolpert and William Macready in 1997 implies that no learning or optimization algorithm that works best on all given problems (Wolpert and Macready, 1997). A classifier can be chosen depending on the type of data at hand. Since then, there had been so many state-of-the-art classifiers that were developed. Some of the most commonly used classifiers are logistic regression, multinomial logistic regression, Naive Bayes classifier, perceptron algorithm, linear discriminant analysis, least squares support vector machines, quadratic classifiers, k-nearest neighbor kernel density estimation, decision trees (random forests), and neural networks.

Many of these classifiers can be categorized as linear classifiers. A classification algorithm is a linear classifier if it uses a linear function or linear predictor that assigns a score to each category k based on the dot product of a weight vector and the feature vector. The linear predictor is given by the score functions, $Score(X_i,k)=\beta_k X_i$, where X_i is the feature vector for the observation i, β_k is the weight vector corresponding category k. Observation i is mapped by the linear predictor to the category k with the highest score function $\beta_k X_i$. Examples of linear classifiers include logistic regression, the perceptron algorithm, support vector machines, and linear discriminant analysis (Yuan et al., 2012).

Data scientists employ techniques and theories drawn from many fields of mathematics, particularly algebraic topology, statistics, information science, and computer science. In Mathematics, in particular, there is a growing field called topological data analysis (TDA). It is an approach that uses tools and techniques from topology to analyze datasets. In the past two decades, TDA has been applied in various areas of science, engineering, medicine, astronomy, image processing, and biophysics.

One of the motivations in TDA is analyzing the shape of data and one of the main tools researchers use is persistent homology (PH). PH is a method for computing topological features of a space of points which persists across multiple resolutions (Carlsson, 2009),(Edelsbrunner and Harer, 2008),(Edelsbrunner and Harer, 2010). It is based on the well-understood algebraic topology where invariant features can be derived algebraically. These gathered invariant features are sensitive to small changes in the input parameters which makes PH attractive to researchers who study qualitative features of data. PH involves representing a point cloud by a filtered sequence of nested complexes, which are turned into novel representations like barcodes and then interpreted statistically and qualitatively based on persistent topological features which were gathered (Otter et al., 2017). A detailed discussion of pertinent information about the homology of simplicial complexes and the process of computing persistent homology of a point cloud can be found in the appendix.

Computation of PH has been applied in various areas including image analysis, shape comparison and recognition, network analysis, computer visions, computational biology, oncology, chemical structures, and many more. Developments in the various aspects of computing PH have been increasing at a very rapid rate. Various software were also developed to provide advanced and beginning practitioners platforms to compute PH or develop new techniques in computing PH. These include JavaPlex, Perseus, Dipha, Dionysus, jHoles, GUDHI, Rivet, Ripser, PHAT, R-TDA, and many more (Otter et al., 2017), (Pun et al., 2018).

This study is focused on the development of a supervised classification algorithm that mainly uses persistent homologies of the datasets to solve classification problems. Persistent homology, which has been around for only a decade has been getting so much attention in the past few years. Published works about the fusion of these topics are quite new. Pun et al. (2018) published a survey of persistent-homology-based machine learning algorithms and their applications. They presented a roadmap on how to use persistent homologies to refine machine learning algorithms such as support vector machines, tree-based methods and artificial neural networks. Their work was the inspiration in this study on how to extract topological features based on persistence barcodes which resulted from

Deleted: which

Deleted: has

Deleted: homology

Deleted: its

Deleted: homology

Deleted:

computing data’s persistent homology. In their study, these features were considered as additional attributes to enhance machine learning algorithms. While in this study, the topological features based on persistence barcodes/diagrams were directly used and the main considerations in the proposed classifiers.

PERSISTENT HOMOLOGY CLASSIFICATION ALGORITHM (PHCA)

The use of persistent homology in topological data analysis has been gaining attraction among researchers and data scientists. PH is mainly used to analyze the shape of a given dataset. A given point cloud undergoes a filtration process which turns it into a sequence of nested simplicial complexes. This is done by considering a finite number of increasing parameters and recording the sublevel sets that track changes in topological information. These changes can be documented in many ways, but the most popular ones are in terms of persistence barcodes or persistence diagrams. From these visualizations, the appearance and disappearances (birth and death) of intrinsic topological features like homology groups and Betti numbers are recorded and interpreted. The persisting duration (life span) of these topological features which are evident in the PH visualizations are essential in analyzing the qualitative and topological properties of data under study. PH has been used also to improve many machine learning algorithms. A list of these instances were mentioned and discussed by Pun et al. (2018). However, one of the main results of this study is the development of a supervised machine learning algorithm that mainly uses persistent homology of sets of data which can be used to solve classification problems.

Given a dataset or a point cloud composed of instances that belong to various classes, the first task is to divide the dataset into a training set and testing. Then, the goal is to analyze the dataset and develop a persistent-homology-based algorithm that will correctly identify the class to which each point in the testing set belongs to.

Consider a point cloud of size M composed of $(n+1)$-dimensional data points. Suppose that in each point, the first n entries are the attributes/features of the given point, and the $(n+1)$-th entry gives the class where the point belongs to. The M points in the dataset are sorted into classes and each of the classes is split into a training set and testing set. For instance, in all the validation runs, we divide each of the classes into at least 80% training set and the remaining points into the testing set. Suppose there are k classes and each class i, $i = 1,2,...,k$, is composed of M_i points. Suppose also that in each class, there are m_i points in the training set and $M_i - m_i$ points in the testing set. If m is the sum of the m_i's, then m is the size of the training set, and $M - m$ is the size of the testing set.

Let X be an $m \times (n+1)$ matrix in which rows represent the points in the training set. Similarly, let Y be an $(M - m) \times (n+1)$ matrix containing the points in the testing set, the testing point cloud. Furthermore, let X_i be an $m_i \times (n+1)$ matrix which contains the training set points belonging to class i. Call each of these matrices as training cloud for class i.

Before commencing the training, set the maximum dimension, denoted by $maxd$, that will be used in forming the Vietoris Rips complex filtration of the point clouds and computing the persistent homology of each of the training clouds. The parameter $maxd$ is usually set to one or two during the validation runs. Validation runs show that these values of $maxd$ are sufficient and the use of larger values of $maxd$ will result to a longer computation time and may not be practical. Furthermore, there is also a need to set the maximum scales, denoted by $maxsc$. The scale here refers to the size of the epsilon balls to be considered in computing the persistent homology and topological features of the dataset. Preferably, the $maxsc$ is set to be half the maximum distance between any two points in the point cloud.

After identifying the point cloud X_i for class i, where i goes from 1 to k and setting $maxd$ and $maxsc$, the algorithm may proceed to the following iterative steps.

Step 1. Training/Learning Stage For each i, $i \in \{1,...,k\}$, form the Vietoris Rips complex filtration for each point cloud X_i for class i. Then, for each i, $i \in \{1,...,k\}$, compute the persistent homology of X_i, based on the Vietoris Rips complex filtration for each point cloud X_i. The result in computing the persistent homology of a point cloud is an $ntf \times 3$ matrix, where ntf is the number of d-dimensional topological features that appear in the filtration. Denote this matrix by $\mathrm{P}(X_i)$.

These topological features include the connected components, the loops, the voids, and so on. The

Deleted: s

Deleted:

Deleted: this

Deleted: which

Deleted: which

Deleted: a

Deleted: is

Deleted: the

Deleted: s

Deleted: Scale

Deleted: the

Deleted: s

number of topological features varies depending on the filtration. Let $tf_{i,j}$ be the j-th topological feature of the point cloud X_i. The first column entries give the dimension of each of the topological feature $tf_{i,j}$. Denote this by $dtf_{i,j}$, where $i = 1,...,k$ and $j = 1,...,ntf$. These entries take the values 0 for connected components, 1 for loops/holes, 2 for voids, and so on. The entries in the second column give the birth time of each of the topological feature $tf_{i,j}$, and the third column entries gives the death time of each $tf_{i,j}$. Denote the birth time and death time of topological feature $tf_{i,j}$ by $\beta_{i,j}$ and $\delta_{i,j}$ respectively. Visual presentation of each of the resulting persistent homology of a given point cloud can be in the form of a persistence barcode or a persistence diagram.

Step 2. Testing/Classification Stage. For each of the $M-m$ data points in the testing set, identify the class/category to which each data point belongs to.

Recall that Y is an $(M-m)\times(n+1)$ matrix, where each row is a data point in the testing set. Let Y_j be the j-th row of Y and the j-th data point in the testing set. Let the first n entries of Y_j be the data point's attributes and the $(n+1)$-th entry be the data point's target class.

For each $j \in \{1,2,...,M-m\}$ and for each $i \in \{1,2,...,k\}$ append Y_j to X_i after the last row of X_i. Name the resulting matrix $XY_{i,j}$. Perform filtration and PH computation on $XY_{i,j}$. That is, compute $\mathrm{P}(XY_{i,j})$. Record the change in topological features from X_i to $XY_{i,j}$. Specifically, record the change from $\mathrm{P}(X_i)$ to $\mathrm{P}(XY_{i,j})$. In this regard, consider two sets of point clouds, say point cloud A and point cloud B. Suppose there is an additional point p, to which we want to classify, whether it belongs to point cloud A or B. The proposed algorithm in this study will perform the classification using topological features based on persistent homology. This technique is different from the techniques used in the existing classifiers. Supposed that point p is closer to point cloud A than point cloud B. Then, the persistent homology of $A\cup\{p\}$ possibly will have more topological features compared to the persistent homology of $B\cup\{p\}$. Also, the birth of new topological features will occur much earlier in $A\cup\{p\}$ and the death of some existing topological features may come earlier in $A\cup\{p\}$.

With this phenomenon in mind, the terms in the score function, which measure the change in topological features from X_i to $XY_{i,j}$, are with reference to the following metrics.

(a) Let $\Omega_{i,j}$ be the difference of the sum of the entries of the first column of $\mathrm{P}(X_i)$ from the sum of the entries of the first column of $\mathrm{P}(XY_{i,j})$.

(b) Let $\Phi_{i,j}$ be the difference of the sum of the entries of the third column of $\mathrm{P}(X_i)$ from the sum of the entries of the third column of $\mathrm{P}(XY_{i,j})$.

(c) Let $\mu\Omega_{i,j}$ be the difference of the mean of the entries of the first column of $\mathrm{P}(X_i)$ from the mean of the entries of the first column of $\mathrm{P}(XY_{i,j})$.

(d) Let $\mu\Phi_{i,j}$ be the difference of the mean of the entries of the third column of $\mathrm{P}(X_i)$ from the mean of the entries of the third column of $\mathrm{P}(XY_{i,j})$.

(e) Let $AM_{i,j}$ be the sum over all k of the absolute value of the difference of the mean of the entries of the k-th column of X_i and the mean of the entries of the k-th column of $XY_{i,j}$.

(f) Let $W_{i,j}$ be p-th Wasserstein distance of $\mathrm{P}(X_i)$ from $\mathrm{P}(XY_{i,j})$, where p is set to 2.

The score function $Score(Y_j,i)$ is computed as

$$Score(Y_j,i) = -\Omega_{i,j} + \Phi_{i,j} - \mu\Omega_{i,j} + \mu\Phi_{i,j} + AM_{i,j} + W_{i,j}$$

Deleted: vary

Deleted: s

Deleted: s

Deleted: s

Finally, each data point Y_j is assigned by the linear predictor to class i with the lowest score function $Score(Y_j, i)$ over all i.

What follows is the pseudo-code for the persistent homology classification algorithm (PHCA).

Deleted: pseudo

EVALUATION METHODOLOGY

Classification is an instance of supervised learning. It is the task of identifying which of the categories a new observation belongs to, based on a training set of data containing observations whose category membership is known. Classifier is the term used to refer to the algorithm that implements the classification and the mathematical function used by the classification algorithm to map an observation to a category. A dataset is composed of $(n+1)$-dimensional data points, whose first n entries are called attributes of the observation and the $(n+1)$-th entry is one of the k categories to which the observation belongs to. The attributes can be real, integer, or categorical. The number of attributes, n, and the number of categories, k, can be any fixed natural numbers. As the number of data points increases, or as the number of attributes increases, the amount of computer time used to solve a classification problem also increases.

Algorithm 1 Persistent Homology Classification Algorithm

Require: $X_1, X_2, \ldots X_k$, Y, $maxd$, and $maxsc$

Ensure: $Class(Y)$ or $Class(Y_j)$ for each j procedure

TRAINING STAGE

$\forall i \in \{1,2,\ldots,k\}$

P$(X_i) \leftarrow (nt\ f)\times 3$ *matrix, a result of computing PH of* X_i end

procedure TESTING STAGE for j = 1 to $M - m$ do for i = 1 to k do

$XY_{ij} \leftarrow X_i \cup \{Y_j\}$

P$(XY_{ij}) \leftarrow (nt\ f)\times 3$ *matrix, a result of computing PH of* XY_{ij}

Compute for $\Omega_{i,j}, \Phi_{i,j}, \mu\Omega_{i,j}, \mu\Phi_{i,j}, AM_{i,j}, W_{i,j}$

$Score(Y_j, i) = -\Omega_{i,j} + \Phi_{i,j} - \mu\Omega_{i,j} + \mu\Phi_{i,j} + AM_{i,j} + W_{i,j}$

$Class(Y_j) \leftarrow \arg\min_{\forall i}\{Score(Y_j, i)\}$

end for

end for

end procedure

Deleted: procedure

The classification algorithm developed in this study was validated by solving a number of classification problems involving various classical validation datasets and a synthetic dataset. It should also be noted that validation of the proposed algorithm in this study was implemented using R and the R-package TDA. The different data used in the validation process were described in the following subsection.

Validation Datasets.

There were four datasets used in validating the proposed PHCA; three classical datasets and one synthetic dataset. The number of classes per dataset is either two or three, while the number of attributes per dataset ranges from two to seven.

1. Iris Plants Dataset

The Iris plant dataset created by Fisher (1936), available at the UCI Machine Learning Repository (Dua and Graff, 2017), retrieved at https://archive.ics.uci.edu/ml/datasets/iris, one of the commonly used dataset in pattern recognition, is composed of 150 observations. The dataset is divided into 3 categories or sub-populations, Iris Setosa, Iris Versicolour, and Iris Virginica. Each category is comprised of 50 data points. All of the 4 attributes of each data point, sepal length, sepal width, petal length, and petal width, are expressed in centimeters.

2. Wheat Seeds Dataset

The wheat seeds dataset was created by Charytanowicz et al. (2010) at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin, available at the UCI Machine Learning Repository (Dua and Graff, 2017), and retrieved at https://archive.ics.uci.edu/ml/datasets/seeds. The dataset is composed of 210 observations which are divided equally into 3 categories: Kama, Rosa, and Canadian wheat variety. That is, there are 70 observations per category. Each data point is characterized by seven attributes: area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient, and length of kernel groove. All of these parameters were real-valued and continuous.

Deleted: is

3. Social Network Ads Dataset

The social network ads dataset was created by Raushan (2017) and retrieved at https://www.kaggle.com/rakeshrau/social-network-ads/version/1. The dataset is composed of 400 data points. The observations were classified into two categories, whether a customer purchased a product (143) or not (257). Each data point has two attributes, age, and estimated salary. This classification task is considered as a bivariate classification problem.

4. Synthetic Dataset

The author created this dataset by generating 200 uniformly sampling points from each of the following figures, the circle defined by $x^2+y^2 = 25$, the sphere defined by $x^2+y^2+z^2 = 1$, and the torus defined by $\left(3-\sqrt{x^2+y^2}\right)^2+z^2=1$. The x,y, and z coordinates of the 600 points served as the attributes, and the category was assigned according to which figure the points belong to.

Performance Measure.

Measure of performance of the proposed PHCA were quantified and then compared with the performance of major classification algorithms with respect to some validation datasets. The metrics used to evaluate the methods were accuracy, sensitivity, and specificity. To compute for these metrics, the respective confusion matrix for each method for the testing set was generated first. A confusion matrix is a table used to describe the performance of a classification model on a set of test data for which the true values are known. The confusion matrix gives the number of data points per class that are correctly predicted or incorrectly predicted.

Deleted: which

For instance, consider a particular class, say C_i, among k classes. Then, we can define the following for each $i \in \{1,2,\ldots,k\}$.

TP_i is the number of true positives in class C_i, or the number of instances in C_i which are predicted to belong in C_i.

TN_i is the number of true negatives in class C_i, or the number of instances outside C_i which are predicted to not belong in C_i.

FP_i is the number of false positives in class C_i, or the number of instances outside C_i which are predicted to belong in C_i.

FN_i is the number of false negatives in class C_i, or the number of instances in C_i which are predicted to not belong in C_i.

The three metrics per class C_i are computed as follows

$$\text{Sensitivity of Class } C_i = \frac{TP_i}{TP_i + FN_i}$$

$$\text{Specificity of Class } C_i = \frac{TN_i}{TN_i + FP_i}$$

$$\text{Accuracy of Class } C_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}$$

A high sensitivity prediction in Class C_i implies that the reliability of predicting that an instance doesn't belong to C_i is high. However, predicting that an instance belongs to C_i with high sensitivity is inconclusive.

On the other hand, the high specificity of prediction in Class C_i implies that the reliability of predicting that an instance belongs to C_i is high. And, predicting that an instance doesn't belong to C_i with high sensitivity is inconclusive.

Validation Procedure.

The following procedure details the steps implemented to measure the performance of PHCA as compared to other classification algorithms. These steps were performed for all of the four datasets.

1. Consider the dataset as a point cloud X. Divide it into 2 parts, training set, and testing set. For all the validation runs, we have split the dataset to at least 80% training set and the remainder to testing set.

2. Solve the classification problem using the proposed PHCA and each of the five algorithms: Linear discriminant analysis (LDA), Classification and Regression Trees (CART), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF). Depending on the algorithm used, utilize the training set and classify each point in the testing set. Information about the nature of these classifiers, including examples and program codes, are available in Subasi (2020), Stanimirova et al. (2013), Breiman et al. (1984), Loh (2011), Neath and Johnson (2010), Cortes and Vapnik (1995), Ho (1995), and Ho (1998).

3. Construct the confusion matrix per classification algorithm.

4. Compute the performance of each classification algorithm in terms of accuracy, and sensitivity, and specificity per class.

RESULTS AND DISCUSSION

Presented here are the performance of the proposed PHCA and the five major classification algorithms in solving four classification problems. Program codes written in R which implements PHCA, LDA, CART, KNN, SVM, and RF can be found on https://github.com/mlddelara/PHCA. There is a section for the discussion of validation results for each of the classification tasks. Presented in each section are the persistence diagrams and the persistence barcodes of the training sets. Recall that PHCA works in a way that a data point in the testing set will be classified under a class if its inclusion in the particular class' training set results to the least change in the persistence diagram or persistence barcode of the training set with the additional data point.

Iris Plants Dataset.

The dataset is comprised of 50 data point from each of the three types of iris plant, namely, Iris Setosa, Iris Versicolour, and Iris Virginica. Each data point is composed of four features and a class label. For each class, ten data points were set aside to be part of the testing set and the remaining forty points were collected as the training set per class.

Figures 1, 2, and 3 shows the persistence diagram and persistence barcode of the respective training sets. These are the representations of computing the persistent homology of each of the training set per class.

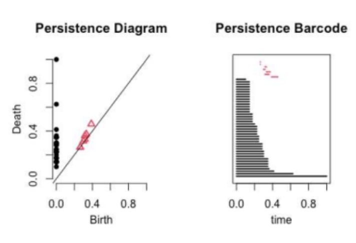

Figure 1. Peristence Diagram and Barcode for the Iris Setosa (Class 1) Training Set

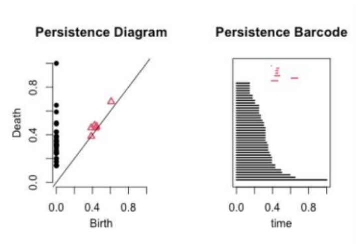

Figure 2. Peristence Diagram and Barcode for Iris Versicolour (Class 2) Training Set

Figure 3. Peristence Diagram and Barcode for Iris Virginica (Class 3) Training Set

Table 1 shows the performance of PHCA and the five major classification algorithms in terms of accuracy, sensitivity per class, and specificity per class. PHCA ranked third in terms of accuracy. That is, of 30 testing data points, only one was wrongly classified. SVM performed equivalently with PHCA, while CART and RF performed poorer with 2 mistakes each. On the other hand, LDA and KNN performed perfectly for this problem.

Classifier	Accuracy	Sensitivity per class	Specificity per class

		Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
LDA	100%	100%	100%	100%	100%	100%	100%
CART	93.33%	100%	100%	80%	100%	90%	100%
KNN	100%	100%	100%	100%	100%	100%	100%
SVM	96.67%	100%	100%	90%	100%	95%	100%
RF	93.33%	100%	100%	80%	100%	90%	100%
PHCA	96.67%	100%	90.91%	100%	100%	100%	95.24%
Number of Data Points:		150		Number of Classes:		3	
Training Set Size:		120		Number of Attributes:		4	
Testing Set Size:		30					

Table 1. Result of classifying the Iris dataset using the six classifiers

Wheat Seeds Dataset.

The dataset is comprised of 70 data points for each of the three types of wheat varieties, namely, Kama, Rosa, and Canadian. Each of the data points has seven attributes and a class label. For each class, there are 14 testing data points and 56 training data points.

The persistence diagram and persistence barcode of the respective training set per class was computed and represented in Fig. 4, Fig. 5, and Fig. 6.

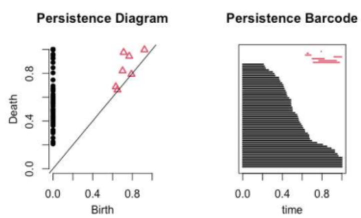

Figure 4. Peristence Diagram and Barcode for Kama Variety (Class 1) Training Set

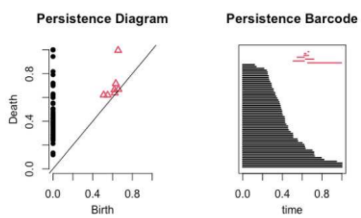

Figure 5. Peristence Diagram and Barcode for Rosa Variety (Class 2) Training Set

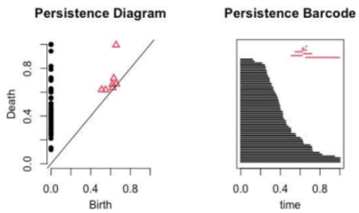

Figure 6. Peristence Diagram and Barcode for Canadian Variety (Class 3) Training Set

Table 2 shows the performance of PHCA and the five major classification algorithms in terms of accuracy, sensitivity per class, and specificity per class. PHCA got the highest accuracy, together with RF and SVM. These algorithms wrongly classified only one data point among 42 testing data points. Moreover, PHCA got the highest sensitivity and specificity for each of the classes. On the other hand, CART performed the worst in terms of accuracy which wrongly classified three data points.

Classifier	Accuracy	Sensitivity per class			Specificity per class		
		Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
LDA	95.24%	92.86%	100%	92.86%	96.43%	100%	96.43%
CART	92.86%	92.86%	100%	85.71%	92.86%	100%	96.43%
KNN	95.24%	92.86%	100%	92.86%	96.43%	100%	96.43%
SVM	97.62%	100%	100%	92.86%	96.43%	100%	100%
RF	97.62%	100%	100%	92.86%	96.43%	100%	100%
PHCA	97.62%	100%	100%	93.33%	96.55%	100%	100%

Number of Data Points:	210	Number of Classes:	3
Training Set Size:	168	Number of Attributes:	7
Testing Set Size:	42		

Table 2. Result of classifying the Wheat Seeds dataset using the six classifiers

Social Network Ads Dataset.

The dataset is comprised of uneven number of observations per class. There are 143 data points for class 1 and 257 data points for class 2. Each of the data points has two attributes and a class label. The former represents observations from customers who purchased a product. There are a total of 80 testing data points and 320 training data points.

The persistence diagram and persistence barcode of the respective training set per class was computed and shown in the Fig. 7 and Fig. 8.

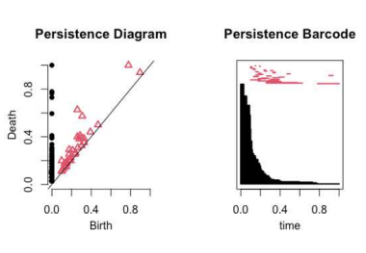

Figure 7. Peristence Diagram and Barcode for Purchaser (Class 1) Training Set

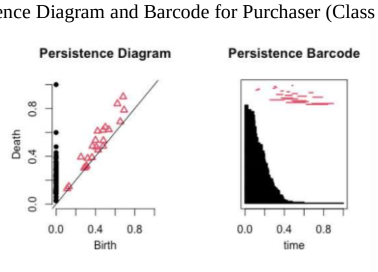

Figure 8. Peristence Diagram and Barcode for Non-purchaser (Class 2) Training Set

Table 3 shows the performance of PHCA and the five major classification algorithms in terms of accuracy, sensitivity, and specificity. PHCA ranked third in terms of accuracy. It got 100% sensitivity, but lower specificity at 90.91%. SVM performed equivalently with PHCA, in terms of accuracy. LDA and KNN got 100% accuracy, but RF and CART got the lowest accuracy of 93.33%.

Classifier	Accuracy	Sensitivity	Specificity
LDA	86.08%	90.20%	78.57%
CART	87.34%	86.27%	89.29%
KNN	82.28%	92.16%	64.29%
SVM	86.08%	88.24%	82.14%
RF	86.08%	90.20%	78.57%
PHCA	82.72%	91.30%	71.43%
Number of Data Points:	400		
Training Set Size:	181		
Testing Set Size:	119		
Number of Classes:	2		
Number of Attributes:	2		

Table 3. Result of classifying the Social Network Ads dataset using the six classifiers

Synthetic Dataset.

The dataset is comprised of 600 data points. There are three classes with 200 data points per class. Each of the data points has three attributes and a class label. For each of the three classes, there are 40 testing data points and 160 training data points.

The persistence diagram and persistence barcode of the respective training set per class was computed and represented in Fig. 9, Fig. 10, and Fig. 11.

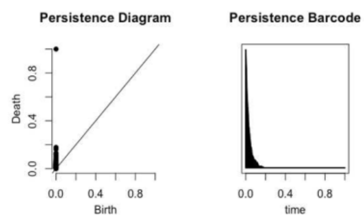

Figure 9. Peristence Diagram and Barcode for Circle (Class 1) Training Set

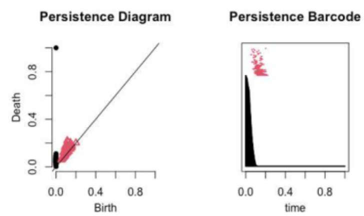

Figure 10. Peristence Diagram and Barcode for Sphere (Class 2) Training Set

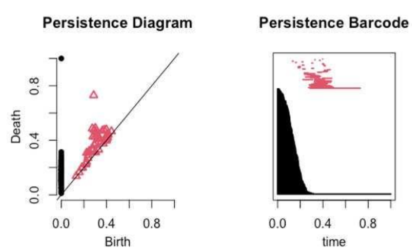

Figure 11. Peristence Diagram and Barcode for Torus (Class 3) Training Set

Table 4 shows the performance of PHCA and the five major classification algorithms in terms of accuracy, sensitivity, and specificity. PHCA, together with CART, KNN, SVM, and RF, performed perfectly with 100% accuracy, sensitivity per class, and specificity per class. While LDA got a low accuracy of 93.33%.

Classifier	Accuracy	Sensitivity per class			Specificity per class		
		Class 1	Class 2	Class 3	Class 1	Class 2	Class 3
LDA	93.33%	100%	82.50%	97.50%	91.25%	98.75%	100%
CART	100%	100%	100%	100%	100%	100%	100%
KNN	100%	100%	100%	100%	100%	100%	100%
SVM	100%	100%	100%	100%	100%	100%	100%
RF	100%	100%	100%	100%	100%	100%	100%
PHCA	100%	100%	100%	100%	100%	100%	100%
Number of Data Points:		600		Number of Classes:		3	
Training Set Size:		480		Number of Attributes:		3	
Number of Data Points:		120					

Table 4. Result of classifying the Synthetic dataset using the six classifiers

Validation of the performance of PHCA was done by comparing its performance in solving four classification problems against the respective performances of the five major classification algorithms in solving the same problems. The four validation datasets are comprised of a varying number of data points, number of classes and number of attributes per observation. In terms of accuracy, sensitivity, and specificity, PHCA and all the benchmark algorithms, excluding LDA, ranked first in two of four validation data sets. However, only PHCA and SVM faired well in all four classification problems. All the other algorithms had the worst accuracy, sensitivity, and specificity in at least one of the problems. CART has the worst performance in solving the Iris dataset and Seeds dataset. LDA has the worst performance in solving the synthetics dataset. And, KNN and RF have the worst performance in solving the Social Network Ads dataset and Iris dataset, respectively.

These validation results do not imply that PHCA is better than any of the other major classification algorithms. But, these results are just evidences to the no free lunch theorem which implies that no learning algorithm works best on all given problems. Moreover, these validation runs imply that PHCA can be at par or even better than some other classifiers in solving some particular classification problems.

What sets PHCA apart from the well-known machine learning classifiers is that it is non-parametric, but at the same time a linear classifier. It is a non-parametric algorithm in the sense that it does not restrict the data to follow a particular distribution nor fix the number of datasets' parameters for the algorithm to work. PHCA works by assigning topological attributes from persistent homology of training data points per classes and uses this as the parameters needed for a linear classifier which the algorithm uses to classify new points. The referred topological attributes include the dimension, birth time, and death time of topological features of the different training datasets and classes, and the Wasserstein distance between classes.

Deleted: is

CONCLUSIONS

The main result of this study was the development of PHCA, a non-parametric but linear classifier which utilizes persistent homology, a major and very powerful TDA tool. Classification tasks are major concerns in the field of machine learning which is why solving these kinds of problems has been a widely studied discipline. The proliferation of the various classification algorithms is further fueled by the fact implied by the no free lunch theorem which implies that there is no single best algorithm which can be used to solve all types of classification problems.

PHCA was validated in this study by using it to solve four different classification problems with varying sizes, number of classes, and number of attributes. PHCA's performance per problem-based on accuracy, sensitivity, and specificity was measured and compared with the performance of five other well-known classifiers. The validation runs show that PHCA can perform well, or even better, than some of the major supervised machine learning classifiers, in solving particular classification tasks. Moreover, this validation activity does not imply that PHCA works better than other machine learning algorithms, but this exposition shows that PHCA can work in solving some classification problems.

Deleted: problem

Validation in this study was limited to relatively small problems which are restricted by the computers used in this study. PHCA can be further validated by considering larger problems and by using more powerful computers which can solve problems with higher dimensions. These future researches could test whether PHCA can still perform at par with or better than other classifiers. Furthermore, various improvements may be imposed on the proposed classification algorithm in this study by considering other topological attributes or by considering persistent homology representations other than barcodes and diagrams. Recent improvements and modifications on the computation of persistent homology may also be adapted to possibly improve the performance of PHCA. PH computations and the validation of the proposed algorithm were implemented using *R* and TDA package in *R*. It should be noted that there are other platforms and solvers which can be used, like JavaPlex, Perseus, Dipha, Dionysus, jHoles, GUDHI, Rivet, Ripser and PHAT, which offer some variations in the the way PH can be computed. Indeed, this study has opened a lot of research opportunities which can be explored by mathematicians, data scientists, topologists, and computer programmers.

ACKNOWLEDGMENTS

This work was created with the guidance of the author's adviser, Dr. Clarisson Rizzie P. Canlubo. The conduct of this research was made possible with the support of the Institute of Mathematical Sciences, College of Arts and Sciences, University of the Philippines Los Baños, and by the Accelerated Science and Technology Human Resource Development Program (ASTHRDP). These institutions had no role in study design, data collection, and analysis, decision to publish, or preparation of the manuscript.

REFERENCES

Alpaydin, E. (2014). *Introduction to Machine Learning, third edition*. Adaptive Computation and Machine Learning series. MIT Press.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Wadsworth and Brooks/Cole Advanced Books and Software.

Carlsson, G. (2009). Topology and data. *Bull Am Math Soc*, 46.

Charytanowicz, M., Niewczas, J., Kulczycki, P., Kowalski, P., Lukasik, S., and Zak, S. (2010). A complete gradient clustering algorithm for features analysis of x-ray images.

Cortes, C. and Vapnik, V. N. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Edelsbrunner, H. and Harer, J. (2008). Persistent homology — a survey.

Edelsbrunner, H. and Harer, J. (2010). *Computational topology: an introduction*. Am. Math. Soc., Providence.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7(Part II):179–188.

Ho, T. K. (14–16 August 1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC,*, page 278–282.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Loh, W. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23.

Neath, R. and Johnson, M. (2010). Discrimination and classification. In Peterson, P., Baker, E., and McGaw, B., editors, *International Encyclopedia of Education (Third Edition)*, pages 135–141. Elsevier, Oxford, third edition edition.

Otter, N., Porter, M. A., Tillmann, U., Grindrod, P., and Harrington, H. A. (2017). A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17.

Pun, C. S., Xia, K., and Lee, S. X. (2018). Persistent-homology-based machine learning and its applica- tions – a survey.

Raushan, R. (2017). Social network ads, version 1.

Stanimirova, I., Daszykowski, M., and Walczak, B. (2013). Robust methods in analysis of multivariate food chemistry data. *Chemometrics in Food Chemistry*, page 315–340.

Subasi, A. (2020). Machine learning techniques. In *Practical Machine Learning for Data Analysis Using Python*, pages 92–202. Academic Press.

Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82.

Yuan, G., Ho, C., and Lin, C. (2012). Recent advances of large-scale linear classification. *Proceedings of the IEEE*, 100(9):2584–2603.