Human Activity Recognition, Topological Data Analysis and R

David Clancy

***Abstract*—This research project set out to use topological data analysis (TDA) on a commonly used human activity recognition data set in the statistical analysis language, R. It was found that R is not well suited to TDA tasks on data sets of any substantial size. This is not due to anything intrinsically related to R but rather due to the implementations used in the two currently available TDA packages. TDA performed reasonably accurately as an unsupervised clustering technique on the human activity recognition data set. In comparison with inspection of principal components analysis results and hierarchical clustering, TDA has similar accuracy to the others while running in about the same time as hierarchical clustering. Using TDA as a classification tool via random forests on the most persistent points from sliding windows from the raw data failed to perform better than rudimentary classification techniques on the main data set.**

I. Introduction

The data for this analysis come from the University of California Irvine's Machine Learning Repository. This repository houses many data sets which can be used on a variety of machine learning tasks such as prediction, clustering, classification, et cetera. The specific data set for this analysis is the Human Activity Recognition Using Smartphones data set[1]. The data were released in 2012 by the original researchers, Anguita, Ghio, and Oneto[2]. These researchers collected the data due to the potential uses in healthcare applications. Since the motions were being captured by a cell phone, the researchers wished to use an effective classification tool which was not too computationally intensive. As such, they designed a support vector machine (SVM) which exploited fixed-point arithmetic in order to reduce the computational load.

The data themselves were collected from 30 subjects. Each subject was asked to perform six activities, walking, walking upstairs (often referred to as as simply "upstairs"), walking downstairs ("downstairs"), standing, sitting, and laying. While performing these activities, the subjects wore a Samsung Galaxy SII on their waist. The Samsung Galaxy SII contains an accelerometer and gyroscope for measuring 3-axial linear acceleration and angular velocity at a constant rate of 50Hz. The subjects were also filmed performing these activities so that true labels could be applied to the data.

Using the raw data from the phones, the researchers applied noise-filtering and sampled in fixed-width sliding windows of 2.56 seconds with 50% overlap. From each of these sliding windows the researchers extracted features such as the means, standard deviations, median absolute deviations, autocovariances, autocorrelations, et cetera from the raw inputs. The researchers released both the cleaned and sampled set as well as the raw data from the phones. The full data set was also split into a training set (70% of the data, 21 subjects) and a test set (30% of the data, 9 subjects).

Their SVM performed as expected and sacrificed only a small amount of accuracy. Table 1 below shows their results in the form of a confusion matrix. Their results brought positive implications for Ambient Intelligent systems for human activity recognition in smartphones. Even today, just a few years later, we see a variety of applications which utilize the phone's accelerometer and gyroscope to count steps, show activity levels, et cetera.

Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	*Recall %*
Walking	**109**	2	3	0	0	0	95.6
Upstairs	1	**98**	37	0	0	0	72.1
Downstairs	15	14	**114**	0	0	0	79.7
Standing	0	5	0	**131**	6	0	92.2
Sitting	0	1	0	3	**108**	0	96.4
Laying	0	0	0	0	0	**142**	100
Precision %	87.2	81.7	74.0	97.8	94.7	100	**89.0**

TABLE I
THE CONFUSION MATRIX FOR THE RESULTS OF THE ORIGINAL RESEARCHERS' SVM. THE TRUE ACTIVITIES ARE THE ROWS AND THE PREDICTED ACTIVITIES ARE THE COLUMNS. THE DIAGONAL ENTRIES ARE THE CASES CORRECTLY CLASSIFIED.

II. Methods

A. TDA in R

R is a free and open source language and environment for statistical computing and graphics. The language has seen rapid growth on both the developer and user sides in recent years. The active development community has written over 5,000 packages for R. As a statistician, the author's first inclination was to find a package implementing TDA and use that to carry out the analysis for the data set. Upon searching the Comprehensive R Archive Network (CRAN), at the time of publishing, there were only two packages which claimed to implement the necessary tools for TDA. The two packages were 'pHom' by Andrew Tausz[3] and 'TDA' by Brittany T. Fasy, Jisu Kim, Fabrizio Lecci, and Clement Maria[4].

Tausz's pHom was first published in February 2014. The package claims to compute persistent homology from point clouds and display persistence plots among other things. Using the code they give for simple examples yields accurate results in a reasonable time. However, every attempt to use this package with synthetic or real data not from the example - no matter how large or small the number of dimensions and

number of cases - yielded terminal errors causing R to crash and reset. Therefore, it was no longer considered an option for this analysis.

Fasy et. al. wrote 'TDA' and first published it in August, 2014. Once again, this package claims to compute persistent homology and persistence diagrams among other TDA related tools. The package worked well with both provided examples and on synthetic and real data. The only reservation with this particular implementation was the speed at which it completed the tasks. This package uses the C++ packages GUDHI and Dionysus for the majority of the computation in computing persistent homology.

In determining where to do the TDA computation, the only reasonable options remaining were R's 'TDA' package and Duke's TDATools in MATLAB. In order to test them a synthetic data set was created (code on GitHub [5]) to test on both. This data set was 600 cases sitting in $\mathbb{R}^6$. There were 6 groups with 100 observations each and they should have separated very well, with only a small amount of overlap. I set the distance bound on both computations to a very large number so that it was essentially getting to the point where every point was connected with every other. In each, I computed the zeroth and first dimensional persistence diagrams. R's implementation took about 53 seconds to run while the MATLAB implementation took about 23 seconds to run; MATLAB was more than twice as fast. Since this was only a data set of a fairly small size while the data set of interest was 7352 cases sitting in $\mathbb{R}^{561}$, MATLAB's implementation was chosen for the data set.

B. TDA Results from Noisy Data

In order to accurately interpret how well TDA does on the data we have, it is important to know what a data set that we know separates well looks like. In order to do that, the synthetic data set from above can be used as a benchmark. The data were generated in such a way that each group should be distinct from the neighboring groups. One easy way to check this is to use principal components analysis (PCA) and visualize the first two components to check the separation between groups. Figure 1 shows the results of this visual check, and it's clear to see that the six groups do separate out quite well.

Armed with the knowledge that the data set separates well - but not extraordinarily so - the results of TDA are the main interest here. Through running this data set in TDATool's rca1pc function, and importing those results back to R, we can see the persistence diagram in Figure 2. As you can see though, with a point cloud the zero dimensional diagram really only contains one dimensional data. Every point is born at time 0 and so all connected components are born at time 0 and the diagram looks like a vertical line. Therefore, from here on out the zero dimensional persistence of point clouds will be shown as histograms as in Figure 3. Also note that because $Persistence = Death - Birth$, the persistence and death of all these points are equal and the x-axis for the histogram is labeled persistence. What we see here is a steep decline on the left side of the histogram before a long stretch of bins with 0 frequency before 6 bins with frequency 1 (the last point actually has infinite persistence, but has been represented at 3). This indicates that these 6 points are representatives of clusters which connect to one another much later than their constituents have connected to one another. In TDA, this is what separated clusters would look like in noisy data.

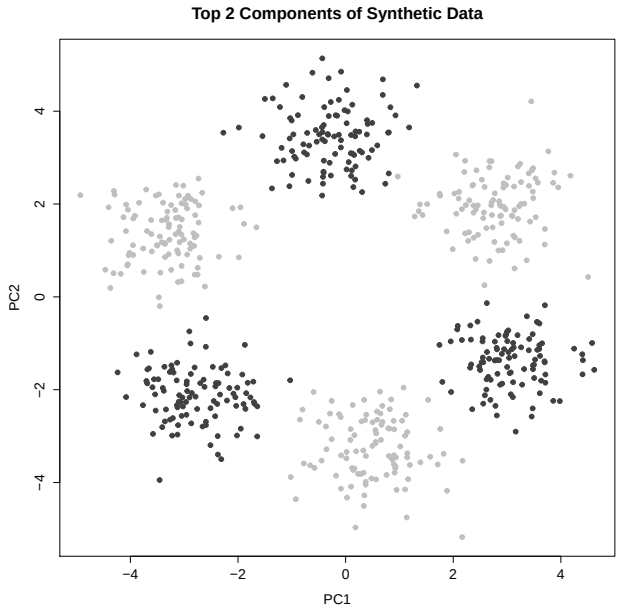

Fig. 1. The top two components of our synthetic data set. There are 6 groups, colored in two shades of grey, alternating (so that the reader can see them when printed in black and white).

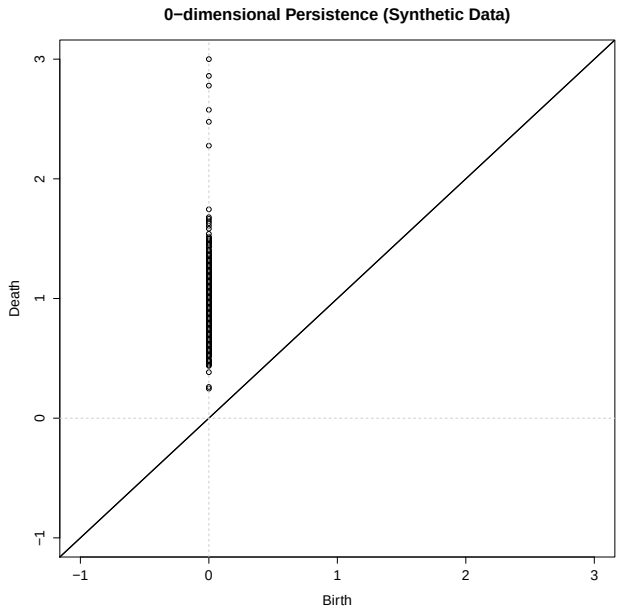

Fig. 2. 0-dimensional persistence diagram for our synthetic data. All connected components are born at time 0. Death time is equal to persistence in this case as well. Also note that the point with a persistence of 3 is actually a point of infinite persistence.

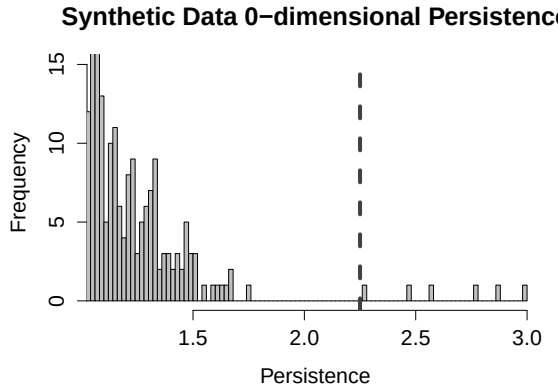

Fig. 3. Histogram of 0-dimensional persistence in our synthetic data set. The dashed line shows the delineation between the 6 most persistent points ('real' clusters) and the rest of the diagram. The x-axis has been truncated here in order to show the right tail in greater detail. Also note that the point with a persistence of 3 is actually a point of infinite persistence.

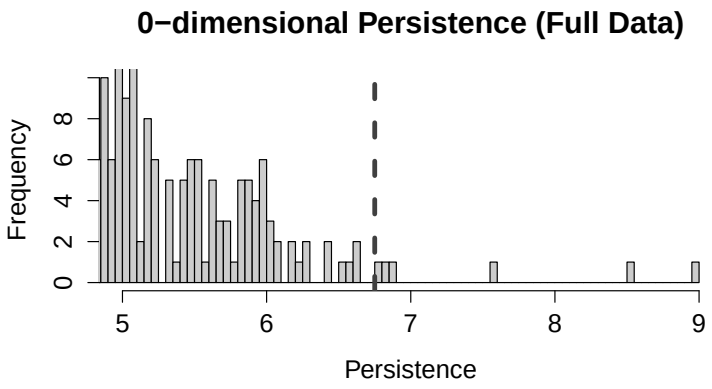

Fig. 4. Histogram of 0-dimensional persistence in our full data set. The dashed line shows the delineation between the 6 most persistent points (number of categories) and the rest of the diagram. The x-axis has been truncated here in order to show the right tail in greater detail. Also note that the point with a persistence of 9 is actually a point of infinite persistence.

III. Clustering

With a benchmark for what good separation looks like and a decision on the toolkit to use, analysis on the data could begin in earnest. The first attempt at analysis used TDATool's rca1pc function with the entire data set. Unfortunately, MATLAB could not complete this code in 30 hours of run time with 13 gigabytes of RAM and without interruption. Since the majority of that time was spent computing first dimensional homology, the rca0pc function was attempted instead. This function was able to complete in less than 10 minutes on the data. The ideal result here would be that there are six points that separate themselves well from the rest of the points. A priori however, from looking at Table 1, it seems like the moving activities (walking, upstairs, downstairs) may not separate out too well. Figure 4 shows the persistence histogram for our full data set.

We see that there are 3 points definitely separated from the rest of the points. However the next 3 most persistent points fit well into the shape of the histogram as it descends towards 0. Treating this as an unsupervised clustering technique (that is, not knowing there were 6 categories beforehand) it would appear to only have 3-4 clusters in the data. This certainly is not the ideal result from our data, however, it fits our a priori estimates fairly well.

A. Comparison

TDA successfully told us there were clusters present in our data set, while the number of clusters was not extremely accurate, it was successful on some level. How does that result stand up to other methods though? The two methods chosen to use for benchmarking are the two that the author was most familiar with at the time, hierarchical clustering and inspection of PCA. The first method, hierarchical clustering takes the original data, chooses the two closest points, notes the distance between them, groups them together, then treats them as one point at their centroid. This process continues until all points are in the same group. The distance between points when they

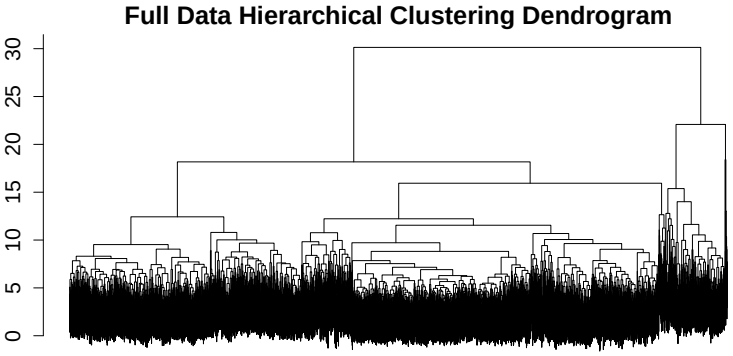

Fig. 5. Dendrogram as the result of hierarchical clustering of the data set. The number of vertical lines crossed at any given height is the number of clusters at that point. This suggests about 4 clusters.

connect can then be put onto a dendrogram. The longer the vertical bar before connection, the further away two groups are. Figure 5 shows the dendrogram for the full data. By inspection, one would probably say about 4 clusters exist in the data. Once again, not the true number of labels, but it seems to concur with the TDA. That's promising for the usefulness of TDA but not so great for the ability to cluster this data set accurately.

The next method for unsupervised clustering is inspection of PCA. Earlier in the paper, we saw how well PCA allows us to visualize separation in a data set even when that separation would require more than two dimensions to see without alteration. Included are two scatterplots which show some of the separation which becomes apparent from the use of PCA. When the plots were considered and delineations made in the cases of clusters appearing, the result was a total of 6 groups. This was the only method to yield the right number of groups, however, it is fairly subjective and another analyst could have ended up with another number. Authors note: It should be noted that I knew there were 6 groups going in and while I

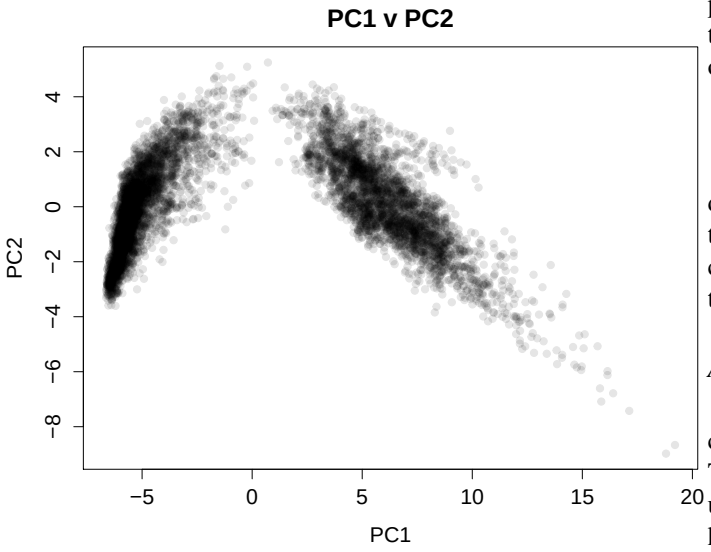

Fig. 6. First two principal components separate very clearly into two clusters (these are actually stationary and moving activities).

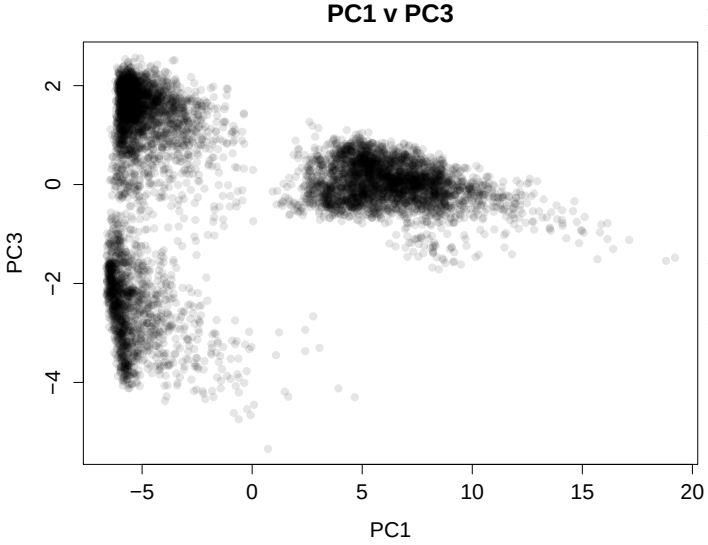

Fig. 7. First and third components separate that left cluster into from Figure 6 into two separate clusters. This separates laying from the other stationary activities.

tried to make sure that didn't affect my decisions, there is a very real chance that it did actually affect my decisions.

All in all, this process took about 2 hours. While this did result in the correct number of clusters, it was a very expensive process in terms of programmer time. In our comparison then, we are left with TDA and hierarchical clustering which performed about evenly and took a reasonable amount of time (for 0-dimensional persistence on the TDA side) and PCA which yielded the correct number of clusters but took much more time. If the analyst wishes to use first dimensional homology for clustering, that becomes a much more computationally expensive proposal. While first dimensional homology may prove to be more useful, a data set of even this size is beyond the capability of a personal computer in a reasonable amount of time.

IV. Classification

Having done unsupervised clustering on our training set data, predicting new data via classification now becomes the focus. The classification done here will be supervised classification, that is we have a test set to which we know the true labels.

A. Direct, TDA Based Methods

Currently, there do not seem to be any approaches for classification which use homology or topology at its core. There are many ways of taking the diagram output and using classification tools on that, but no tools specifically homological in nature. Therefore, two shall be proposed here. The first is a naive approach which can be tested currently; the second is a more complex approach which cannot be tested at this time due to a lack of implementation.

The first method, henceforth referred to as the naive approach, would be to classify a point in the same class as the point it connects to when it dies. The existing point which it connects to may already be part of a larger connected component, but the only thing the algorithm cares about is the classification of the single existing point. The point which kills the connected component of our single new observation will always be the closest point to the new observation. Because of this fact, this approach is the same as a k-nearest neighbors approach with k set equal to 1. Because of this equivalence, this algorithm was tested and the results are shown in Table 2. We see that the overall accuracy was 87.9% which was very close to the SVM the researchers used. For being such a simple method, the naive approach seems to be quite accurate.

Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	*Recall %*
Walking	**473**	31	53	0	0	0	84.9
Upstairs	8	**422**	46	0	0	0	88.7
Downstairs	15	18	**321**	0	0	0	90.7
Standing	0	0	0	**389**	81	3	82.2
Sitting	0	0	0	99	**451**	1	81.9
Laying	0	0	0	1	0	**533**	99.8
Precision %	95.4	89.6	76.4	79.2	84.8	99.2	**87.9**

TABLE II
THE CONFUSION MATRIX FOR THE RESULTS OF THE NAIVE APPROACH TO TDA CLASSIFICATION. THE TRUE ACTIVITIES ARE THE ROWS AND THE PREDICTED ACTIVITIES ARE THE COLUMNS. THE DIAGONAL ENTRIES ARE THE CASES CORRECTLY CLASSIFIED.

The second method, henceforth referred to as the complex approach, does not necessarily rely on a single vote for the classification. With this method, when the connected component of the new single observation dies, the constituents of the connected component it is joining are queried and the new observation is assigned the most frequent class within the connected component. This approach does have limitations. For instance if there is unequal sample sizes between classes

and a new observation is further away from its class than that class is from a neighboring one, the new observation will be classified incorrectly. However, since this method uses a group voting system, this method seems to be more robust than the naive approach and could yield more accurate results.

B. Using Raw Data

Because each observation in the main data set was essentially the summary statistics and information about a 2.56 second window of time, the raw data were used for first dimensional TDA. Going back and using the raw data meant that first dimensional homology may be able to pick up on loops while walking upstairs that could distinguish it from walking downstairs better than the previous methods could. It is unclear whether or not the raw data provided from UCI had already had noise filtering on it, but no additional modifications were made for this analysis. The raw data consisted of nine features plus subject and activity identifiers (Python code for going from the UCI data to a single .csv also found on GitHub).

The same 2.56 second sliding windows with 50% overlap were used for our first dimensional TDA. From the raw data, a first dimensional persistence diagram was computed for each window, then the persistences from the 30 most persistent first dimensional loops were pulled and stored in a data set along with the subject and activity. This data set was then put through a random forest procedure for classification. Random forest was chosen because a recent analysis of classification techniques on a large number of UCI's data sets revealed that random forest is one of the best procedures across the board[6] and so it would give us the highest likelihood of picking up on things in this data from TDA. The results, however, were much worse than the results from the other prediction techniques. Table 3 shows the confusion matrix for this classification.

Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	*Recall %*
Walking	**400**	57	39	0	0	0	80.7
Upstairs	75	**211**	142	0	1	0	49.2
Downstairs	47	100	**248**	0	0	0	62.8
Standing	0	0	0	**195**	156	171	37.3
Sitting	1	0	0	103	**297**	157	53.2
Laying	0	0	0	138	182	**254**	44.3
Precision %	76.5	57.4	57.8	44.7	46.7	.436	**54.0**

TABLE III
THE CONFUSION MATRIX FOR THE RESULTS OF THE RANDOM FOREST ON THE 30 MOST PERSISTENT POINTS FROM THE FIXED WITH SLIDING WINDOWS. THE TRUE ACTIVITIES ARE THE ROWS AND THE PREDICTED ACTIVITIES ARE THE COLUMNS. THE DIAGONAL ENTRIES ARE THE CASES CORRECTLY CLASSIFIED.

It should be obvious that across the board this prediction method is the worst of any tried or discussed in this paper. There could be many reasons for this. One such reason is that the raw data has not had any noise filtering. If this is the case, the TDA calculations here could be picking up more noise than it can handle. It could be that the movement loops which seem obvious when walking may not come through at the waist. It could be that another time interval may better pick up the motions, though a 2.56 second should allow for multiple steps to be taken.

V. CONCLUSIONS

This paper has discussed TDA, R, and their usage on a Human Activity Recognition data set at some length. At the moment R is lacking a good implementation of TDA which runs in a reasonable time frame for all but very small data sets. This is unfortunate since TDA appears to be a fairly good unsupervised clustering tool. Statisticians such as the author who primarily use R would find an implementation very useful as opposed to needing to transfer between MATLAB and R via .csv files. It was also found that on data sets of the magnitude used in this paper, 1-dimensional homology is unrealistic for computation on a personal computer. On the classification front, TDA had little to offer in this data set. However, this does not mean it is a bad classification tool in general. It has been used very successfully in other applications. In this application though, TDA seemed to fail on the raw data and do slightly worse than the original researcher's SVM when using the naive approach to classification directly using TDA (and not the diagrams).

A. Ideas for Future Work

In the future, it would be interesting to how the complex classification method described earlier performs on both synthetic and real data sets. An implementation of that algorithm would not be too computationally intensive, and should still be reasonable for data sets considerably larger than this one. The one dimensional persistence data from the full data set (not the raw data) might hold useful information about classification or clustering on the activities, however, given the fairly long width of the sliding windows, I doubt it would improve the accuracy rate a considerable amount. Finally, it might be interesting to use the 1-dimensional persistence diagrams (either raw data or full data set) in order to identify different subjects through clustering and classification.

ACKNOWLEDGMENT

The author would like to thank the original researchers for making their data available, University of California Irvine for creating and maintaining the Machine Learning Repository and Paul Bendich for teaching a great computational topology class.

REFERENCES

[1] U. o. C. Irvine, "Human Activity Recognition Using Smartphones Data Set," 2012. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphone

[2] D. Anguita, A. Ghio, and L. Oneto, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," *International Workshop of Ambient Assisted Living (IWAAL)*, 2012. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-35395-6_30

[3] A. Tausz, "Package phom ," 2014. [Online]. Available: http://cran.r-project.org/web/packages/phom/phom.pdf

[4] B. T. Fasy, J. Kim, F. Lecci, and C. Maria, "Introduction to the R package TDA," 2014. [Online]. Available: http://cran.r-project.org/web/packages/TDA/vignettes/article.pdf

[5] D. Clancy, “Computational Topology Fall 2014 Scripts,” 2014. [Online]. Available: https://github.com/DJC37/compTDA2014F
[6] M. Fernandez-Delgado, E. Cernadas, S. Barro, and D. Amorim, “Do we Need Hundreds of Classifiers to Solve Real World Classification Problems,” pp. 3133–3181, 2014. [Online]. Available: http://jmlr.csail.mit.edu/papers/volume15/delgado14a/delgado14a.pdf