Topological Data Analysis:
3D Image Clustering with Persistent Landscape Vectors

Leonard J. Strnad, Corinne M. Sandor

2121 Euclid Avenue, Cleveland, OH 44115

Topology, Dr. Bubenik – Cleveland State University

ljstrnadiii@yahoo.com, c.m.sandor@vikes.csuohio.edu,

Abstract. *Topological data analysis is an emerging field in Statistics and Machine Learning. JavaPlex* [1] *is a friendly persistent homology computation library for Matlab and Java and is used to construct explicit filtered simplicial complexes to find their corresponding persistent homologies. Given a triangulation of various 3D images, we filter the triangulation by the angle between neighboring triangles, calculate the Persistent Landscape Vectors, and plot the corresponding first two principal components. Significant clustering occurs which suggests the persistent landscape vector may be used to classify 3D images.*

1. Introduction

Given a triangulation of various 3D images, the goal is to see if we could classify or cluster each figure type by their corresponding persistent homology. To do this, we focus on the defining features of each figure i.e. places of tight concavity between triangles.

A persistent homology introduces new simplices to the complex at different time steps according to our filtration function. Our function is based on angles between neighboring triangles. Since neighboring triangles in tighter concave areas tend to have larger angles between them, we have our function set up so that at each time step a certain amount of triangles appear according to angles where triangles with larger angles appear first. In computing the persistent homology, we obtain a corresponding persistent vector space or homology group at each time which can be described by a set of barcode intervals. Each equivalence class in the persistent homology has a barcode interval which represents the class's birth and death.

Barcode intervals can also be represented as a matrix which has a corresponding persistence landscape vector. Each element in the persistence landscape vector corresponds to a radius centered at time t determined by the maximum number of k intersecting barcode intervals [2]. The persistence landscape vector may have very high dimension, therefore, we can use principal component analysis to reduce the dimension. Reducing the dimension allows simpler analysis of the persistence landscape vectors.

2. Data

The data we are using is distributed by the TOSCA (Toolbox for Surface Comparison and Analysis) project; Nonrigid world 3d database [1]. There are 148 unique nonrigid 3D

[1] https://github.com/appliedtopology/javaplex

figures which include 9 cats, 11 dogs, 6 centaurs, 6 seahorses, 3 wolves, 17 horses, 15 lions, 21 gorillas, 1 shark, 24 female figures, and two different male figures (15 poses for one male and 20 poses for the other). In the database, we are provided with .png images of each figure along with two other text files: one containing a list of XYZ vertex coordinates (.vert) and the other containing a list of triangular faces (.tri) which triangulates the corresponding figure.

3. JavaPlex and Persistent Homology Computation

In order to compute the persistent homology of any given figure, we first use JavaPlex to build a corresponding explicit simplicial complex. A filtered explicit simplicial complex allows us to build a complex at each time interval. We have the option to bring certain simplices in at certain times and by certain filtrations such as height, density, or even angles. We choose to filter our persistent homology by angles between neighboring triangles.

Since we already have a triangulation of each figure, we simply add an interval of triangles at some time step. However, we want to ensure that when a triangle is added its corresponding vertices are added too, so we call the “ensuring all faces” function. The ensuring all faces functions ensures that all subsimplices corresponding to a simplex are included as well.

Once we build each complex, we use the JavaPlex persistent homology algorithm to obtain the corresponding barcode matrix of each figure which we then pass into the persistent landscape vector function [2].

4. Filtering 3D Images by Angle

In order to sort the triangles (in the .tri file) by angle, we use the ”neighbors” function (a method in the triangulation class in Matlab) to find the neighboring triangles for each triangle. We then record the angle between each triangle and all pairs of neighboring triangles into a matrix we call ”theta” and sort by descending order. From there, we partition ”theta” into intervals determined by the number of partitions within a certain percentile. The triangles are added at each interval treating each interval as ”time”. We then compute the persistent homology, find the corresponding barcode intervals, and compute the persistence landscape vector for that image.

Figures 1, 2, 3 demonstrate how the unique characteristics of the 3D images are captured as a result of filtrating by angle.

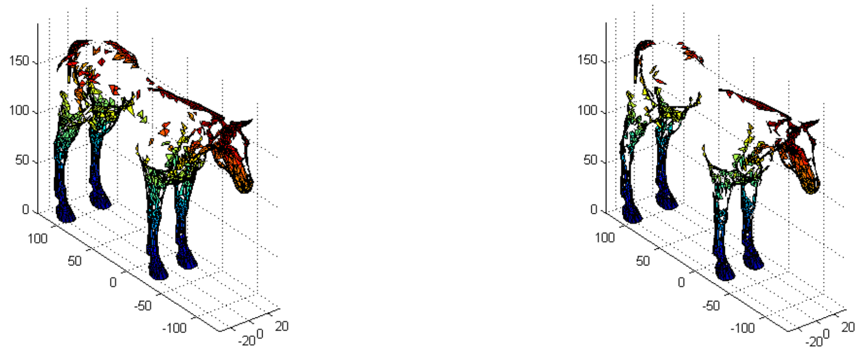


Figure 1. Horse filtered to 70th and 80th percentile

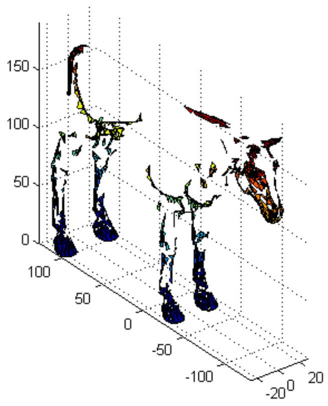
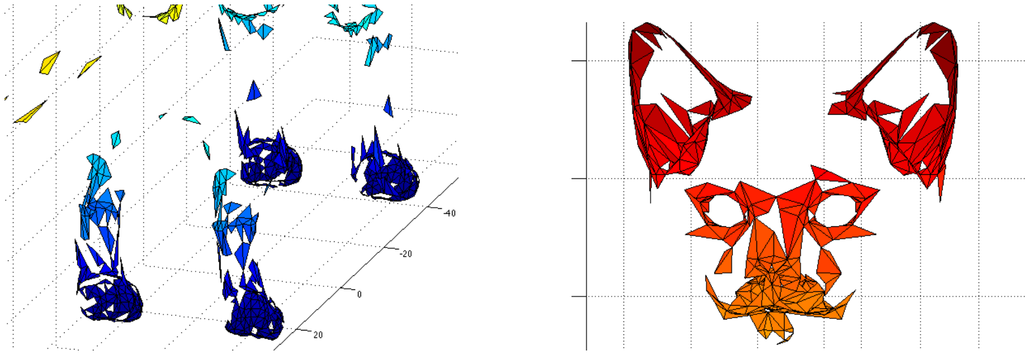


Figure 2. Horse filtered to 90th percentile



(a) legs and paws of cat

(b) face of cat

Figure 3. Filtration of the 90th percentile reveals unique characteristics of the 3D image

Figures 1 and 2 display the image, ”horse0”, at certain percentiles. Note: as the percentile increases, the less the image is ”filled in”. When we specify that we want to filter to the 90th percentile, we are specifying that we want to display the top 10% of the triangles in the ”theta” matrix (i.e. Since ”theta” is sorted by descending angles, the 90th percentile is the top 10% of triangles by angle between neighbors).

Figure 3 demonstrates how the unique characteristics in the 3D images have tighter concavity.

5. Principal Components and the Persistent Landscape Vectors

The persistent landscape vector is a vector of high dimension. At every time step we are considering k overlapping barcode intervals. The persistent landscape vector we are considering has 40 time-steps and 15 overlapping intervals. The dimension of our persistent landscape vectors is (number of time-steps $+1$ $\times$ number of overlapping intervals) $= 615$. In order to reduce the dimension of this vector and retain the information that is contained in them, we consider the corresponding principal components.

A principal component is a linear combination of certain dimensions of the underlying distribution. This linear combination is the result of the dot product of the eigenvectors of the corresponding correlation matrix and the random vector $\mathbf{X}$.

Let's consider the persistent landscape vector as a random vector $\mathbf{X} = (X_1, ..., X_d)^T$ where each entry is a random variable from the same probability space and d is the dimension of the persistent landscape vector. This distribution has a corresponding covariance-variance matrix , Σ, that is determined by $\mathrm{Var}[\mathbf{X}] = \mathrm{E}[(\mathbf{X} - \mathrm{E}[\mathbf{X}])(\mathbf{X} - \mathrm{E}[\mathbf{X}])^T]$. If we first standardize the variables, then the covariance-variance matrix will be the corresponding correlation matrix of the data. Note the correlation and covariance-variance matrix is non-singular and symmetric. Thus, there exists an eigenbasis.

The eigenbasis of this this correlation matrix provides a geometrical basis of the variance from the underlying distribution of this random vector $\mathbf{X}$. The principal components are defined by

$$\mathbf{y_{(i)}} = \mathbf{e}^{\mathrm{T}}_{\mathbf{(i)}} \cdot \mathbf{X} \tag{1}$$

where (i) signifies the principal component, $\mathbf{y_{(i)}}$, is ordered. The principal components are ranked by their corresponding variance. The variance of the principal component is $\mathrm{Var}[\mathbf{y_{(i)}}] = \lambda_{(i)}$ where $\lambda_{(i)}$ is the eigenvalue that corresponds to the eigenvector, $\mathbf{e_{(i)}}$. The proportion of variance explained by the first two principal components can be given by $\frac{\lambda_{(1)}+\lambda_{(2)}}{\sum \lambda}$, where the denominator is the sum of all eigenvalues which is d – the dimension of $\mathbf{X}$ for the correlation matrix.

The principal components allow us to represent the variance of the underlying distribution in a much lower dimension if the proportion of variance explained is significant. For example, if the first two principal components capture 90% of the variance, then we can represent the data in only 2-dimensions.

6. Results

After determining the persistent landscape vectors for each image using $k = 15$ and number of time-steps as 40, we found the correlation matrix of this data and its principal components. It turns out that the first two components explain about 93% of the variance. Thus, we were able to perform some very simple graphical analysis of the biplot of the first two principal components.

The center was calculated for each image type i.e. all horse images, all dog images, etc. We then connected each image instance to its corresponding center and colored all the images of a certain type a unique color to highlight any clustering in the biplot of the principal components.

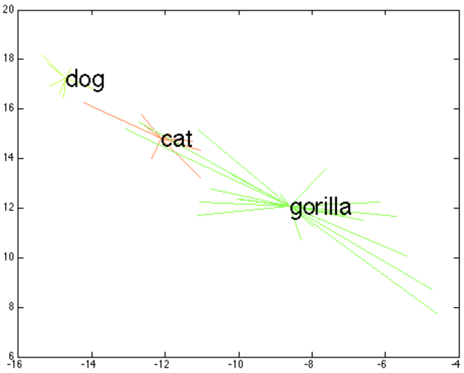


Figure 4. This plot suggests dog, cat and gorilla are distinguishable

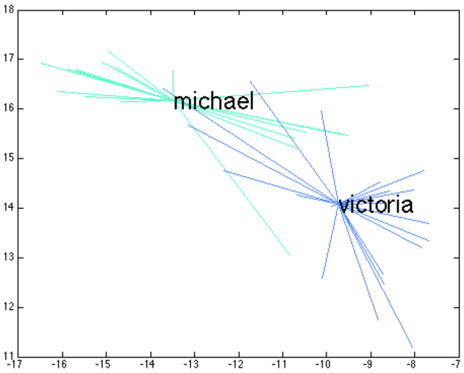


Figure 5. This plot suggests male and female are distinguishable

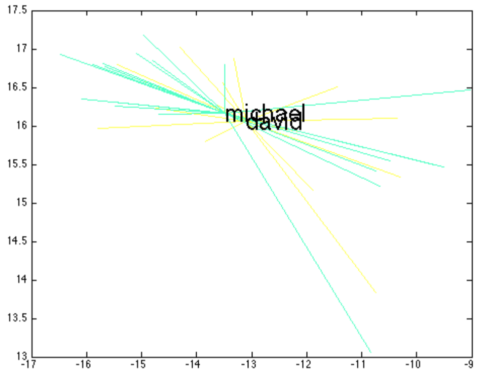


Figure 6. This biplot illustrates the similarity between two males

The plots above suggest the persistent landscape vectors and their corresponding principal components are useful in revealing similarities and differences between 3D non-rigid images. Figure 4 demonstrates that there are significant differences between a dog, cat, or gorilla. Figure 5 suggests that persistent landscape vectors can be used to distinguish male and female images. Figure 6 suggest they can also be used to reveal similarities between images.

The clustering suggests that a classification model could be built where the input space is the random vector $\mathbf{X}$, the persistent landcape vector. The most interesting thing to note is that this classification model would probably be robust to the objects in various positions. All the images of a certain type are in a different pose. This persistent landscape vector appears to be robust to orientation and even the pose or position of the object in the image. Often image classification models are not robust to these conditions and have a lot of pre-processing to do before any analysis.

Further exploration of classification or clustering using the persistence landscape vector and filtering by angle may include different values of k and more or less times steps. We only considered a range of angle values to consider–a percentile–for computational simplicity. Perhaps the persistent landscape vector characterizing the addition of all simplices to complete the triangulation would provide more information and be able to more clearly distinguish image types. Additional study may include trying to classify each image of a particular type by pose.

Overall, the persistent landscape vector seems to contain important information regarding the geometrical structure of the point cloud data that constructs these 3D images. It would be interesting to study what types of inherent properties/information the persistence landscape vector can capture.

References

[1] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008.

[2] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *arXiv preprint arXiv:1207.6437*, 2012.