Using TDA on Art

Carmen Cox

December 12, 2014

1 Introduction

The rich complexity seen woven throughout the history of art was what first gave me the idea to use topological data analysis, or TDA, on images. My hope was that elements of the distinct styles of different works of art, such as the delicate shadowing used by Leonardo Da Vinvi and the strong lines by Pablo Picasso, could be detected using the Rips Complex on the created structure of a simplicial complex on the image. The results of my project gave implications that, indeed, zero and one-dimensional persistence correspond with defining features for a work of art. However, many questions also arose on what it is exactly that TDA is picking up on, along with the potential for future directions of this project.

2 Method and Analysis

2.1 The Dataset: Images

Images are broken up into pixels, each of which has an associated value between 0 and 255 that correspond with the intensity of the color of that particular pixel. A normal, colored image can be thought of as three matrices, where each matrix represents the red, green and blue values of the image. However, in order to simplify the dataset for my project, I used grayscale images instead. A grayscale representation of an image has only one matrix for the grayscale values instead of the red, green and blue values, and each pixel is represented by a float between 0 and 1. Black corresponds with the value 0 and white with 1.

2.2 Creating a Simplicial Complex

I used the code for the Rips Complex, rca1mfscm, to analyze the images I gathered. However, in order to do this, I first had to give the structure of a simplicial complex to the image so that it could be fed into the code. In order to do this, I first assigned a vertex to each pixel, or the corresponding element in the matrix. The value of the vertex was the grayscale intensity value assigned to the pixel while the vertices were numbered according to the indices of the matrix.

Edges were created between the vertices. Specifically, edges were drawn between each pair of horizontally adjacent vertices and each pair of vertically

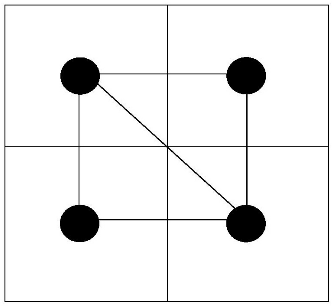

Figure 1: Example of simplicial complex on a small image made up of four pixels.

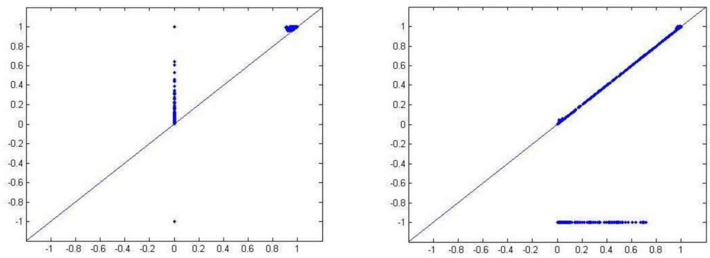

Figure 2: Zero and one-dimensional persistence diagrams for an image of four black dots.

adjacent vertices, so that the image was essentially turned into a grid. However, the Rips Complex kills cycles by the formation of triangles, not squares, so I also drew an arbitrary diagonal in each square of four vertices, as shown in Figure 1. The value assigned to each edge was the higher grayscale intensity of the associated vertices.

Finally, I used a distance bound of 1.2 so that rca1mfscm would run on the entire range of grayscale values from 0 to 1.

2.3 Some Simple Examples

I first ran some examples to see if rca1mfscm would be able to detect at least very obvious zero and one-dimensional cycles of simple designs. First, I used an image that had a white background and four black dots on it. The corresponding zero and one dimensional persistence diagrams are shown in figure 2. While the one-dimensional persistence diagram reflects the fact there there are no one-cycles in the image, the zero-dimensional persistence diagram has a curiously large dot of persistence, with its birth at 0 and its death at 1. Recall that the value of 0 in an element of a matrix corresponds with the color of black on a pixel and a value of 1 corresponds with the color of white. The large dot represents the four black dots coming in at zero, and thus reflecting four different components, but these components are merged together when the white background fills in.

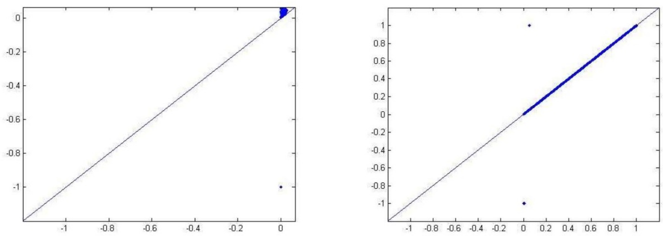

Figure 3: Zero and one-dimensional persistence diagrams for an image of a black outline of a triangle against a white background.

Next, I ran rca1mfscm on the picture of a non-filled-in black triangle on a white background. The resulting dimensional diagrams can be seen in figure 3. There is not much to say about the zero-dimensinoal diagram, which should make sense since there is only one object in the image, that is the triangle. A component is born almost immediately when the black triangle comes in and does not die since the white background grows adjacent to it. However, there is a distinctly large dot in the one-dimensioal persistence diagram, which is what we would hope for since this is a clear one-cycle. But upon deeper reflection, this is also a little peculiar. The cycle begins at 0, when the black triangle appears, and dies at 1 when the white background emerges. However, the white background does not just fill in the inside of the outline of the triangle, which would obviously kill the one-cycle, but also appears on the outside of the triangle as well. It seems that this would change the affect of being able to detect a clear one-cycle but that does not appear to be the case. While it may be tempting to celebrate the reflection of a one-cycle in the persistence diagram of this image, it is also necessary to be suspicious of why this seems to occur so well despite the problem of the surrounding white background. Questions such as these will be summarized at the end and lead to mullings over possible future directions.

2.4 TDA on Art

I ran TDA on multiple works of art by Leonardo Da Vinci and Pablo Picasso. I chose these two artists in particular because of how distinct their styles are, as well as how much their art differ from one another. I also wanted to pick artists that the reader could readily recognize.

While I did not have a chance to run machine learning on the persistence diagrams I produced, I did find some clear and interesting differences just from looking at them. Of course, there is a severe flaw in a dependence on this rough human analysis, but this also provides encouragement for future research into the topic.

I present Da Vinci's Mona Lisa, one of the most well-known paintings in the world, and Picasso's Two Girls Reading as the examples to show the contrasting features TDA was able to pick up on. These paintings are shown in figure 4. Note that I used grayscale images of these paintings rather than their originals colored forms.

Figures 5 and 6 show the produced zero and one-dimensional persistence

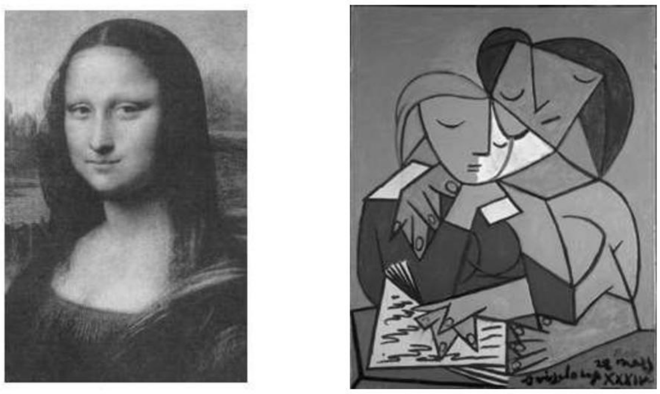

Figure 4: Mona Lisa by Leonardo Da Vinci and Two Girls Reading by Pablo Picasso.

diagrams, respectively. The zero-dimensional persistence diagrams are not terribly different, although there are more higher persistence dots for Two Girls Reading. I believe this correlates with the blocks of color in Picasso's painting in contrast to Da Vinci's more realistic use of gradual changes in color. Perhaps what is most interesting though, are the two immortal dots that occur for Mona Lisa. I used a distance bounds that is higher than the greatest value for any grayscale image, which is one, so all components besides the overall component of the image itself, should have died. The same goes for the Two Girls Reading. This brings up yet another shortcoming in the understanding of what TDA is picking up on with a dataset of images.

The contrasts in the one-dimensional persistence diagrams lends a bigger hint to the differences in the two paintings. The persistence diagram for Two Girls Reading has more dots of high persistence than that of Mona Lisa. Further, there are more dots of low persistence for the Mona Lisa. These two diagrams appear to reflect the more distinct one-cycles found in Picasso's Two Girls Reading. Notice in figure 4 how Picasso breaks up his painting into polygons: a square for a woman's head, a diamond for a neck, a triangle for hair. The boldly lined painting can be thought of as being broken up into closed one-dimensional shapes, or one-cycles. In contrast, the Mona Lisa has no lines, except for the subtle ones that are created naturally between light and shadow.

3 Questions and Future Directions

A natural next step for this project would be to use machine learning. One could take many samples of paintings from different art movements, create persistence diagrams, extract feature vectors and build a classifier. While I only used paintings from two particularly distinct artists, it is quite possible to run TDA on a much wider range of artists. The categorization potentials could also be extended to art movements, different mediums and other qualities of art. Even further, one does not need to be limited to classical art; indeed, per-

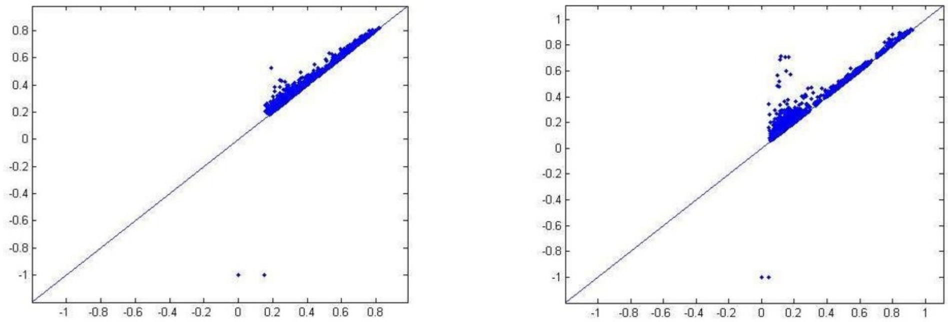

Figure 5: Zero-dimensional persistence diagrams for Mona Lisa and Two Girls Painting, respectively.

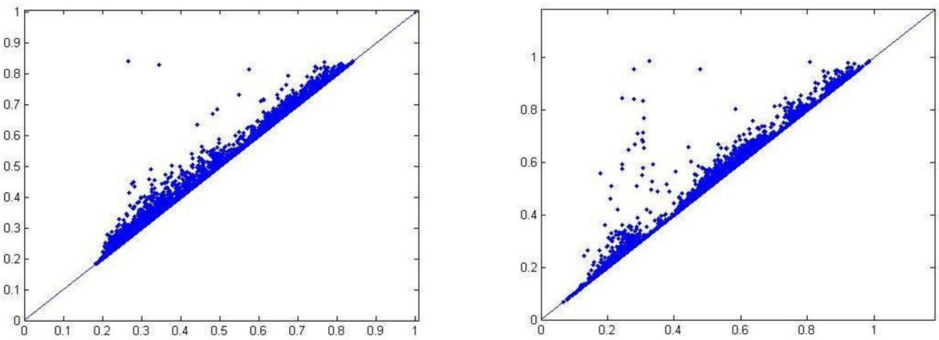

Figure 6: One-dimensional persistence diagrams for Mona Lisa and Two Girls Painting, respectively.

haps TDA could be used on any sort of image such as photographs. Machine learning could help explore the question of whether a computer is able to tell the difference between images and how different must the images be for this to work.

In my project, I only used grayscale images so growth in this project could include figuring out a way to use colored images instead. It may be possible to simply create simplicial complexes for each of the three RGB matrices and concatonate them, although then there is a potential problem of the size of the simplicial complex when inputting it into rca1mfscm.

My biggest problem while running the code was precisely that. Many of the images were simply too big for rca1mfscm to run efficiently on their simplicial complexes and I had to crop many of my images just to be able to produce the persistence diagrams. This also leads to the concern that perhaps different sized images may have an influence on the differences of the resulting persistence diagrams.

Lastly, there is simply a lot going on in between the input of the simlicial complex of the image and the output of the persistence diagrams. The diagrams for the simple examples, and even for the paintings, had instances where some of their dots did not make immediate sense. A more rigorous analysis of images of different zero and one-cycles, perhaps with different grayscale values as well, would be necessary to better understand the what it is exactly that TDA is picking up on.