Computational topology: Image classification using Vietoris-Rips complex

May 13, 2016

Neža Mramor Kosta, Gregor Jerše

Andrej Dolenc, Peter Us, Rok Ivanšek

Contents

Project description **3**

Obtaining and preprocessing data **3**

The classification model **4**
Relation to single linkage clustering algorithm 5
Computational complexity 6

Results **7**
Cup, paper, pen image set 7
General classification 9

Summary **10**

Workload **11**

Figure 1: Sample images from used dataset.

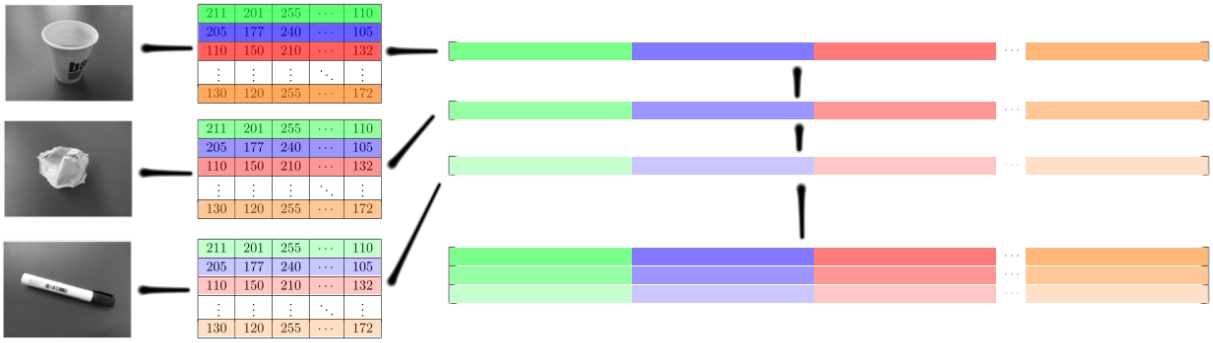

Figure 2: Converting a set of images into one large matrix for use with our model.

Project description

As the title implies, the idea behind the project is to use the Vietoris-Rips complex (from here on VR-cx) for image classification. We took pictures of three generic objects and preprocessed them so we obtained their vector representations. The vector representations of images can then be looked on as points in some n-dimensional space $\mathbb{R}^n$. We build the VR-cx over these points for two different parameters. We use the complexes as a classification model for the images and classify them. We also perform several interesting tests on the generated VR-cx.

Obtaining and preprocessing data

For our dataset we used grayscale photographs of 3 different objects: a coffee cup, crumpled up paper and a pen. Each of the objects was photographed 10 times from slightly different angles and under different illumination. The sizes of the images were 1080 pixels in width and 810 pixels in height. A selection of these photographs can be seen in figure 1.
We can think of images as matrices of size $(a \times b)$ where each entry represents the intensity of the corresponding

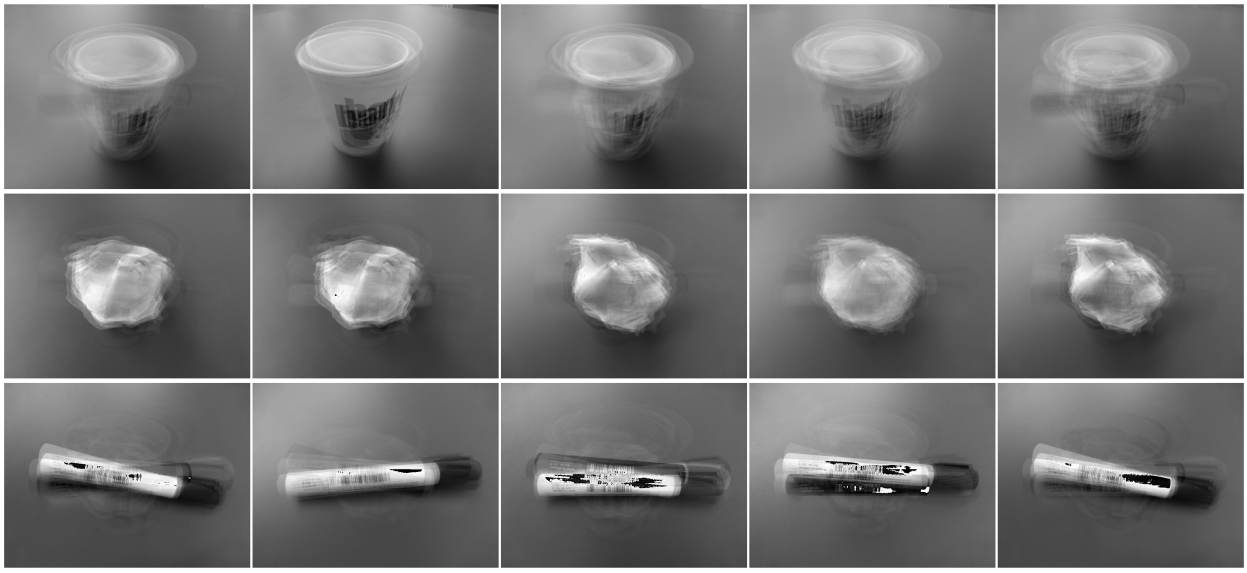

Figure 3: Images from figure 1 after applying preprocessing steps to them.

pixel (meaning in our case we had 30 matrices of size 1080×810). With images in grayscale color space, each of the values is an integer between 0 (black) and 255 (white). In order to prepare the dataset for evaluation with our classification model, we flattened each image's matrix into one long vector of size nm (first row of the matrix being first b elements, second row the next b and so on a times). We then stacked all individual vectors vertically to form a large matrix with $n = 30$ rows, one for each individual image, and ab columns, one for each individual pixel. Figure 2 shows this process graphically, which may help to properly undestand it.

Because we wanted to help our model as much as possible via preprocessing, we scaled/transformed the values of each pixel either using a standard scaler, or by subtracting mean and scaling values to the $[-1, 1]$ interval. The main motivation behind this was to remove varying illumination from our dataset.

With this however we are still left with a very large matrix, so in order to reduce processing time, decrease redundancy and amount of noise, we apply principal component analysis (PCA for short) to it. With this we can to reduce the number of columns from ab to but a few, which makes the dataset much more manageable. Figure 3 shows same sample of images as in figure 1 after applying both scaling and principal component analysis (and inverting both operations).

The classification model

Definition 1 (Vietoris-Rips complex)**.** *Let X be a set of m-dimensional points $X \in \mathbb{R}^m$ and let d be a metric. Pick a parameter $r > 0$. Construct a simplicial complex as follows:*

- *Add a 0-simplex for each point in X.*
- *For $x_1, x_2 \in X$ add a 1-simplex between x_1, x_2 if $d(x_1, x_2) \leq r$.*
- *For $x_1, x_2, x_3 \in X$ add a 2-simplex with vertices x_1, x_2, x_3 if $d(x_1, x_2), d(x_1, x_3), d(x_2, x_3) \leq r$.*
- *...*
- *For $x_1, x_2, ..., x_m \in X$, add a $(m-1)$-simplex with vertices $x_1, x_2, ..., x_m$ if $d(x_i, x_j) \leq r$ for $0 \leq i, j \leq m$; that is, if all the points are within a distance of r from each other.*

The simplicial complex is called the Vietoris-Rips complex and is denoted $VR_r(X)$.

Definition 2. *We say that disjoint subsets* $A_1...A_k$ *of vertices* V *in graph* $G(V, E)$ *are* k *connected components of the graph* G *if the following is true:*

1. *The vertices inside* A_i *are connected i.e. there exists a path between arbitrary two vertices* $a, b \in A_i$, *for every* $i \in 1...k$.
2. *The sets of vertices* $A_1, ..., A_k$ *are disconnected i.e. there isn't an edges* $e(a, b)$ *between a pair of two points* (a, b) *such that* $a \in A_i, b \in A_j, i \neq j$.

Given the input of n m-dimensional points $X \in \mathbb{R}^m$ and a metric d (in our case the eulidian distance) we build a VR-cx for parameter $r = r_2$, where r_2 is the biggest r such that the VR-cx has three connected components. We find the parameter r_2 using a binary search on a set of all possible values of r. We then perform the classification, simply by saying that all the objects in some connected component belong to the same class. Seeing as we have three connected components in our VR-cx we will get three distinct classes.

It does not take much thinking to see that our classification model will produce the exact same results regardless of whether we only use simplices of dimension 1, or if we used all the simplices up to dimension h, where h is an arbitrary number in the range $h \in 1...m$. This follows from Definition 1.

Relation to single linkage clustering algorithm

Our intuition tells us, that the model we build using the VR-cx to classify the images, produces the same results as the well known single linkage clustering algorithm. In this section we aim to prove or at least give a strong intuition that this is indeed the case.
Both algorithms take a set X of n samples with m features as input. We can think of a sample x in the set X as a point in the m-dimensional space $x \in \mathbb{R}^m$. The algortihms then constructs a graph with points X as the vertices (V) and edges E. The k connected components in the constructed graph corespond to the classes of samples.

Single linkage algorithm. The algorithm starts with n connected components (no edges in the graph). n each step the algorithm chooses the two connected components that are closest to each according to some distance metric d (in our case the euclidean distance) and joins them into one by adding an edge between their closest two vertices.

Definition 3. *The distance* D *between two connected components* A *and* B *is defined as the distance of the pair of vertices (one from* A *and one from* B*) that are closest to each other. More formally*

$$D(A, B) = \min_{a \in A, b \in B} d(a, b).$$

The algorithm stops when there are only k connected components left.

Vietoris-Rips classification algorithm. The algorithm builds a (1-dimensional) Vietoris-Rips complex $V_r(X)$ with parameter r. We choose the biggest r such that the VR-cx $V_r(X)$ has k connected components.

To prove that the two algorithms indeed produce the same connected components we will first prove the next claim.

Claim 1. *Let* $G_{sl}(V, E_{sl})$ *be a graph produced by the single linkage algorithm for finding* k *clusters and let* d_{max} *denote the distance between vertices in graph* G_{sl} *that were connected in the last iteration of the algorithm. Graph* G_{sl} *has connected components* $A_1...A_k$. *The graph* $G_{vr}(V, E_{vr})$ *induced by the Vietoris-Rips complex* $VR_{d_{max}}(V)$ *has the same connected components* $A_1...A_k$.

Proof. We prove the Claim 1 by induction on the steps in the single linkage algorithm. We start with a set of vertices V. Let j denote the step (iteration) of the algorithm, e_j the edge added in j-th step and d_j its length. We claim that at each j the graph G_{sl}^j constructed by the algorithm up to that point, has the exact same conected components as G_{vr}^j, that is the graph induced by the Vietoris-Rips complex $VR_{d_j}(V)$.

Base case. For $j = 0$ this is obvious, since this is the initial state of the algorithm. Both graphs G_{sl}^0 and G_{vr}^0 consist only of vertices V. For $j = 1$ the algorithm adds the smallest edge e_1 out of all possible candidates and builds a graph G_{sl}^1. Edge e_1 has length d_1. It is obvious that $VR_{d_1}(V)$ will induce a graph G_{vr}^1 that will also only contain edge e_1, since no other pairwise distance between vertices V is smaller.

Induction step. Here we show that if for some j our claim holds, it will also hold after another iteration of the algorithm i.e. for $j+1$. In $(j+1)$-th iteration, the algortihm finds the edge e_{j+1} with length d_{j+1} and adds it to the graph. By the definition of the algortihm e_{j+1} is the smallest such edge that connects (joins) two seperate connected components. This means that every other edge e' with length $d' < d_{j+1}$ would not join connected components, but would instead just connect two vertices, that are both allready in the same connected component. From the definition of the Vietoris-Rips complex we can see that in the graph G_{vr}^{j+1} there will only be one new edge that will join two seperate connected components, and that will be exactly edge e_{j+1}. All the other extra edges that will be added in G_{vr}^{j+1}, but do not appear in G_{sl}^{j+1} have length less than d_{j+1} and will therefore only connect vertices inside of allready existing connected components of the graph G_{vr}^j. Since by our induction hypothesis graphs G_{sl}^j and G_{vr}^j had the same connected components and we joined two of the same connected components in both graphs, this means that the graphs G_{sl}^{j+1} and G_{vr}^{j+1} also have the same connected components.

We have proven that the graph G_{sl}^j constructed in j-th iteration of the single linkage algorithm indeed contains the same connected commponents as the graph G_{vr}^j induced by $VR_{d_j}(V)$ for an arbitrary j. This also prooves Claim 1. □

Using Claim 1 we see that the connected components in G_{vr} and G_{sl} are indeed the same. We need to take into account that the Vietrois-Rips algorithm takes the biggest such r, so that the graph has k connected components, so $r > d_{max}$. But we can quickly see that the extra edges in the graph induced by $V_r(V_{sl})$ will not change the connected components. After all we allready have k connected components in G_{vr}. To join any two together would mean a violation of a fundemental rule of the algorithm.

Computational complexity

Let us now consider the computational complexity of our model. Since we are only interested in Vietoris-Rips complexes VR_r with simplices of dimension 1, the simplest approach to construct such complex requires us to check the distance between every pair of vertices $(v_1, v_2) \in V \times V$, adding such pair to the final complex if the distance $d(v_1, v_2) \leq r$. In worst case the algorithm would have to return all distinct pairs of vertices, meaning construction of $VR_r(V)$ requires $O(n^2)$ time and consumes $O(n^2)$ space, where n is the number of vertices in V.
The problem is that we don't know the appropriate value for the parameter r. Recall that we are interested in finding biggest r, such that the Vietoris-Rips complex VR_r has as many connected components as there are distinct classes of images. Let r_{max} denote the largest distance between two vertices from X. Note that r we are looking for will always be bounded on the interval $[0, r_{max}]$, and it will furthermore be exactly one of the distances between some pair of vertices. Thus we only have n^2 different possible values of r we need to check, and if we sort them by size and use binary search to find the right one, we can do it in $O(n^2 \log n^2 + \log n^2) = O(n^2 \log n)$ time and $O(n^2)$ space. To count the number of connected components obtained with every different VR complexes, we can use a `union-find` algorithm, which at worst adds $O(n^2)$ time to each run.

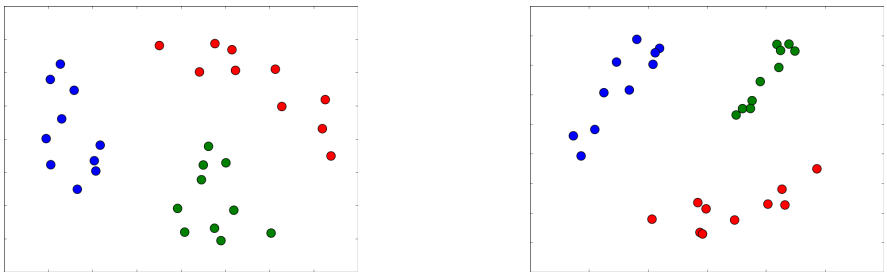

Figure 4: MDS plots visualizing cup (blue), paper (red), and pen (green) samples from the dataset. Left figure is the original dataset. Right figure shows the scaled and PCA transformed dataset. Euclidean distance metric was used.

With this the final time complexity of our approach is $O(n^2 \log n^2 + (n^2 + n^2) \log n^2) = O(n^2 \log n)$ using $O(n^2)$ space. Contrast this with the computational complexity of single-linkage clustering, which with a clever implementation can in optimal case produce solution in $O(n^2)$ time and $O(n)$ space [2].

Results

Cup, paper, pen image set

Effect of preprocessing. We were interested in observing the effect of the preprocessing methods on the samples of our dataset. For this purpose we visualized the data-points using multidimensional scaling technique (MDS) [1]. MDS is a means of visualizing the level of similarity of individual cases from N-dimensional space, to a lower, M dimensional space (in our case: $M = 2$), while preserving between-object distances as well as possible. Therefore MDS plots are well suited for giving intuition of how separable data samples of different classes are. The obtained plots are shown in figure 4. We notice that in both cases the points can be clearly separated into three clusters. We can also observe that points within clusters seem to lie closer together in the preprocessed plot than in the non preprocessed one. It should be mentioned that the position of points in a MDS plot is randomized, and that only distances between points should be observed. Therefore the reader should not confuse the different positions of each of the clusters in the plots as an effect of the preprocessing, since the reason for it is merely a property of the MDS method.

Image classification. In this section we present the results of Vietoris-Rips classification algorithm described in the previous section.

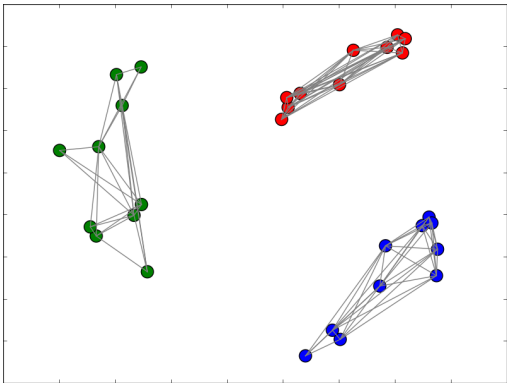

Figure 5: Visualisation of the VR-cx on the cup, paper, and pen dataset.

Initially we tested if the model is able to distinguish the images of cup, paper and pen. The images were preprocessed with scaling between values of $[-1, 1]$, as described in section Obtaining and preprocessing data, and a PCA transformation with 0.75 explained variance. The VR-cx obtained is shown in fig 5. The model had no problems distinguishing between the three classes of images. Each one of the three connected components obtained was belonging to one of the three classes of images. This is not surprising as the three classes of points are clearly separable.

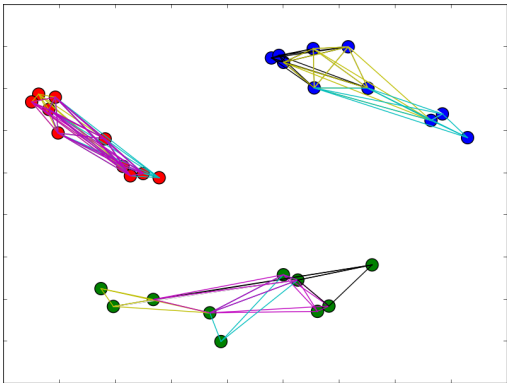

Figure 6: Visualisation of the principal simplicies obtained by VR-cx on the cup, paper, and pen dataset.

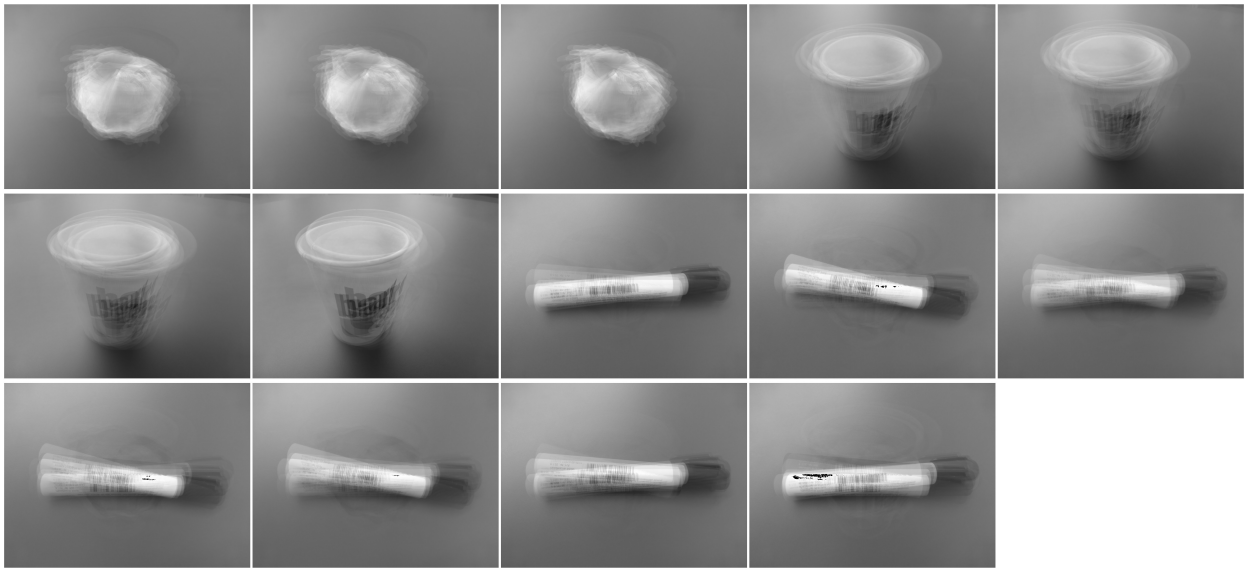

Figure 7: Images representing inverse-tarnsformations of baycenter points of each of the principal simplices ordered by decreasing sizes of simplices.

Principal simplices. In constructed VR-cx we observed 14 principal simplicies (i.e. simplicies which are not faces of any bigger complex), which are shown in figure 6. We can notice that each of the principal simplicies is connecting points within same classes. In figure 7 the images representing barycenteer points of each of the principal simplices are presented. It is interesting to see that there are as many principal simplicies connecting "pen"-points, as there are "paper" and "cup" principal simplicies all together. The intuition for this is that since pen is a "pointy" object, small rotations of the image cause bigger changes in the image than if the object is rounder (which is the case with paper and cup images).

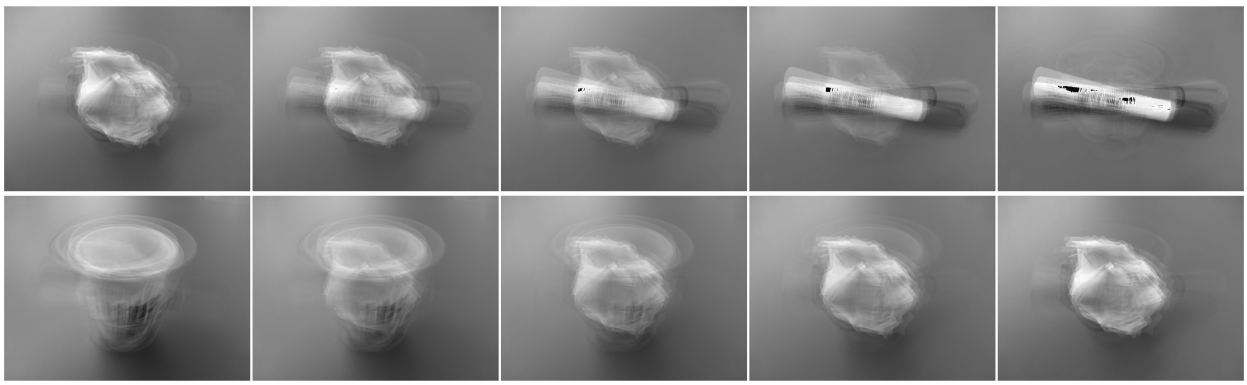

Figure 8: Reconstructed images along various points on 2 edges that join 3 connected components into 1.

Intercluster edges. We were also interested in finding out what images that lie between connected componets look like. To this end, figure 8 shows reconstructed images that would lie on shortest two edges which would join all 3 clusters into one, at distances 0%, 25%, 50%, 75% and 100% along the edges. Note that leftmost and rightmost images are actually images from our dataset (after applying preprocessing steps), while the rest are all completely artificial. Despite this, they still very much resemble just taking affine combinations of values of corresponding pixels. With this we conclude that the images at 50% do not capture the difference between two connected components, but rather their similarity.

General classification

Comparing performance with single linkage clustering. We have shown in section Relation to single linkage clustering, that single-linkage clustering produces the exact same results as our topological clustering method using VR-cx. In addition to the proof we plotted the graphs built by both models to demonstrate this. The connected components of both graphs can be seen on figure 9. We can see the connected components are indeed the same and in this case they corespond to correct clusters.
The strong connection beetween the two algorithms however also implies, that our classification model will face the same problems as single linkage, in other words, it will fail on all datasets where the clusters are not as clearly seperated and/or contain outliers. To demonstrate this we ran both algorithms on one other "bad" dataset where the objects we photographed were not centered, nor were the pictures always shot from the same distance. The graphs build by the two algorithms can be seen on figure 11. It is clearly visible that the classification is bad. If we think a bit further we can also easily see, how outliers in our dataset will always cause the algortihm to fail in correctly classifiying the images. Some samples from this dataset can be seen in figure 10.

Using our model with other datasets. Of course our model can be used to classify (or cluster) datasets with completely different attributes. To prove the point, we tried it on the Iris dataset, which consists of 3 classes with 50 instances each, each class being a different type of Iris plant. There are 4 numeric attributes in total: sepal width and length of the flower, and its petal width and length (all in centimeters). One of the classes is linearly separable from the other two, while the other two are very intertwined.
To verify that our model in fact works, we first ran it on Iris dataset and told it to simply find 2 clusters, which it successfully managed to do (it clustered the dataset into exactly the linearly separated class and the intertwined one). Telling it to find 3 clusters however did not work so well: while it was still able to properly cluster one class, it made a cluster out of both remaining ones and found a single "outlier" sample that was slightly further away from other samples.

Modifying algorithm to prefer simplices of larger dimension. Because we proved that our basic

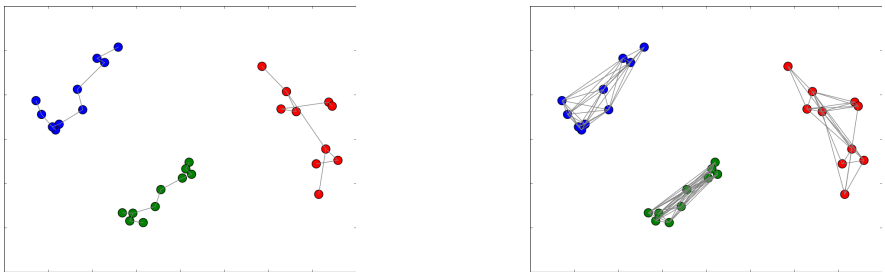

Figure 9: MDS plots visualizing the graphs build by both the single linkage clustering (left) and topological clustering using VR-cx (right) on our dataset.

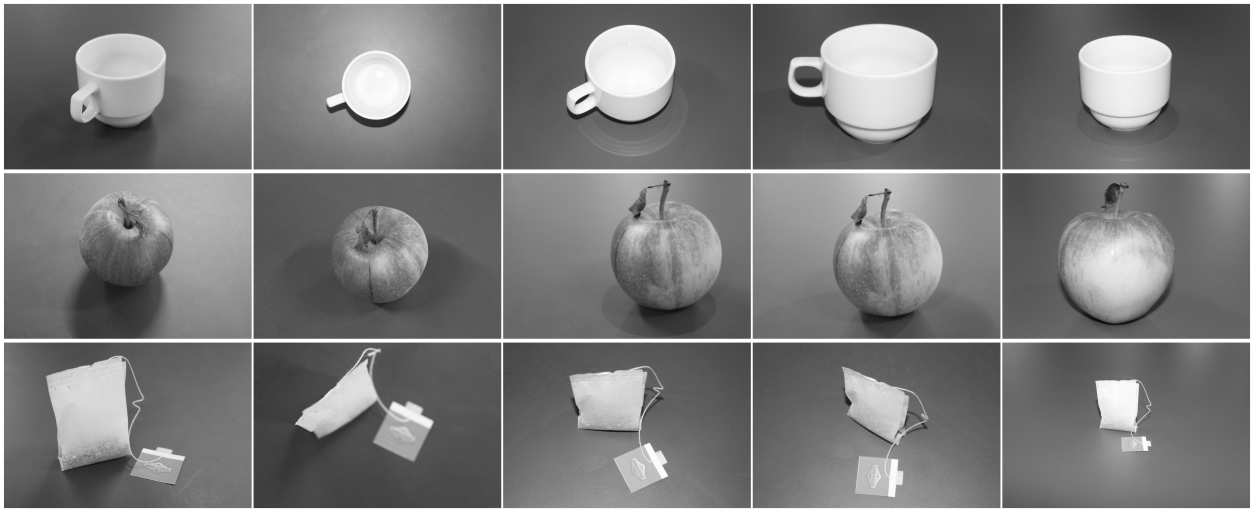

Figure 10: Sample images from the "bad" dataset.

model that only considers simplices of dimension 1 produces identical results as single-linkage clustering algorithm, and because VR cx can produce simplices of higher dimensions, we wondered if we could somehow use those to increase robustness of our algorithm. The main idea is that if some vertices are present in a simplex of higher dimension, they are more likely to be a part of the same class (since all the points are bunched up closer together). Unfortunately this turned out to be useless in practice, as even simplices of higher dimension are still built based on some distance metric between points, and if two clusters of different classes are close enough together, they may form a simplex of very large dimension resulting in a incorrect classification. The idea also would not work too well on thin, long clusters as seen in figure 4. On top of everything, this would also add a second parameter (optimal dimension in addition to r), that the model has to balance out to produce desired number of clusters, and we would have to construct simplices of dimensions ≥ 1 at each run of VR algorithm, further increasing the computational complexity with potentially no benefit.

Summary

Classification of images using Vietoris-Rips complex is equivalent to one of the most basic hierarchical clustering algorithms, single-linkage clustering. It thus has the very same problems and is highly dependent on good preprocessing of the dataset into well separated groups. At the same time, it has a significantly worse performance in both time and space requirements.
While additional intercluster connections that VR cx constructs are interesting from various viewpoints,

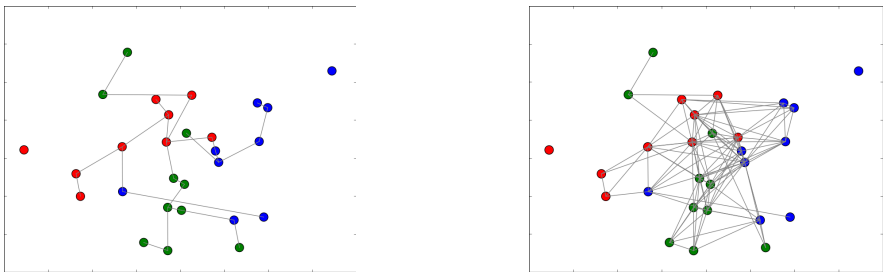

Figure 11: MDS plots visualizing the graphs build by both the single linkage clustering (left) and topological clustering using VR-cx (right) on "bad" dataset.

we fail to see any practical value from them, especially in relation to our problem. We also are not sure about the usefulness of higher dimensional simplices, predominantly given how quickly they can increase time complexity and make the model unusable.

Workload

Our project workload was split among the members fairly. We met biweekly and mostly worked together, sharing ideas and findings. To give a brief overview of what each member did:

- P prepared preprocessing class (StandardScaler, PCA) and wrote a nice abstraction for VR-cx from Dionysus library, added MDS visualizations which were used to reason about the dataset, and played around testing different scalers to see what effect they have on separability of classes;
- R took the photographs, wrote a formal proof on the connection of our model to single-linkage clustering, helped with the implemetation of the final model and implemented step-by-step visualisation of building the model which was very useful in building our understanding of how it works and why it works the way it does; and finally
- A implemented methods for loading and saving images, converted original images to grayscale, resized them and turned them into a dataset that could be used the model, and finally implemented the the model itself.

The analysis of the model and final results were a joint effort, as was writing this final report. A more detailed overview of each member's workload can be seen at github repository of our project: `https://github.com/ptrus/vietoris_rips_image_classification/commits/master`.

References

[1] Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005.

[2] John C Gower and GJS Ross. Minimum spanning trees and single linkage cluster analysis. *Applied statistics*, pages 54–64, 1969.