

Computational topology: Image classification using Vietoris-Rips complex

May 12, 2016

Neža Mramor Kosta, Gregor Jerše

Andrej Dolenc, Peter Us, Rok Ivanšek

Contents

Project description	3
Obtaining and preprocessing data	3
The classification model	3
Relation to single linkage clustering algorithm	3
Computational complexity	5
Results	5
Summary	5

Project description

As the title implies, the idea behind the project is to use the Vietoris-Rips complex for image classification.

TODO: Finish it up

Obtaining and preprocessing data

TODO: Describe our dataset (and show them)

TODO: Describe the process of generating matrix X

TODO: Describe the preprocessing steps and the idea behind it

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 $x_1, x_2, \dots, x_m \in X$, add a $(m-1)$ -simplex with vertices x_1, x_2, \dots, 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 $VR_r(X)$.

TODO: Describe how VR cx is then used to classify images

TODO: Explain why we only care about simplices of dimension 1

Relation to single linkage clustering algorithm

Our intuition tells us, that the model we build using the Vietoris-Rips complex 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.

Definition 2. *We say that disjoint subsets $A_1 \dots A_k$ 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 $i \in 1 \dots k$.*
2. *The sets of vertices A_1, \dots, A_k are disconnected i.e. there isn't an edges $e(a, b)$ between a pair of two points (a, b) such that $a \in A_i, b \in A_j, i \neq j$.*

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 algorithms then constructs a graph with points X as the vertices (V) and edges E . The k connected components in the constructed graph correspond to the classes of samples.

Single linkage algorithm. The algorithm starts with n connected components (no edges in the graph). In 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 Vietoris-Rips complex $V_r(X)$ 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 $A_1 \dots A_k$. The graph $G_{vr}(V, E_{vr})$ induced by the Vietoris-Rips complex $VR_{d_{max}}(V)$ has the same connected components $A_1 \dots A_k$.

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 connected 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 algorithm finds the edge e_{j+1} with length d_{j+1} and adds it to the graph. By the definition of the algorithm e_{j+1} is the smallest such edge that connects (joins) two separate 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 already 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 separate 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 already 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 components as the graph G_{vr}^j induced by $VR_{d_j}(V)$ for an arbitrary j . This also proves Claim 1. \square

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 Vietoris-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 already have k connected components in G_{vr} . To join any two together would mean a violation of a fundamental 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 $(x_1, x_2) \in X \times X$, adding such pair to the final complex if the distance $d(x_1, x_2) \leq r$. In worst case the algorithm would have to return all distinct pairs of vertices, meaning construction of $VR_r(X)$ requires $O(n^2)$ time and consumes $O(n^2)$ space, where n is the number of vertices in X .

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 to check, and if we sort them by size and use binary search to find the right one, we can do it in $O(n \log n)$ time and $O(n^2)$ space. To count the number of connected components obtained with each of different VR complexes, we can use a union-find algorithm, which roughly adds a $O(n^2)$ time to each run.

With this the final time complexity of our approach is $O((n^2 + n^2) \log n^2) = O(n^2 \log n)$ using $O(n^2)$ space.

TODO: Double check the time complexity of union-find

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. **TODO: Find reference from wikipedia.**

Results

TODO: Explain we get same results as with S-L clustering, and with that the same problems as S-L clustering.

TODO: Present MDS graph of our dataset after applying preprocessing, and emphasize different classes/recognized clusters.

TODO: Explain that simplices of larger dimension are just about useless

TODO: Explain what datasets our solution actually works with (e.g. determining position of objects on image)

TODO: Extra tests: images from midpoints of edges, barycenters of simplices, ...

Summary

TODO: Brief summary

TODO: Further work (if there even is any), what we didn't explore.

References

- [1] Leslie Lamport, *L^AT_EX: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.