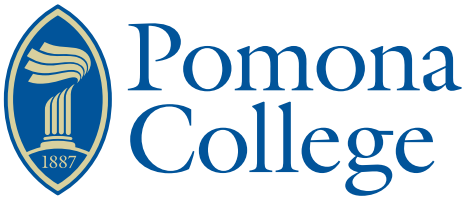

Senior Thesis in Mathematics

Thesis: Topological Data Analysis and Mesh Recognition

Author:
Ra-Zakee Muhammad

Advisor:
Gabriel Chandler

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

June 19, 2022

Contents

0.1 Introduction

MeshLab is a software that can read topological data to form Meshes or topological shapes approximating real life subjects in various positions. These meshes are triangulated surface approximations meaning that the data that is read by MeshLab comes in the form of 3 item tuples of numeric triples (x,y,z) this data creates triangles that when linked together estimate the shape. These meshes, along with a feature descriptor called the heat kernel signature, are important to understanding Sun, Ovsjanikox, and Guibas's 2009 paper A Concise and probably informative Multi-Scale Signature Based on on Heat Diffusion, which discusses how heat diffusion across a surface can provide information about the shape of the object the surface is of.

0.2 Preliminaries: Laplacian

Mode and anti-mode

First we think about scalar valued functions as functions that output either real or complex scalar values. We need the notion of a differentiable scalar valued function or a scalar field f in order to then think about the gradient of $f : \mathbb{R}^n \to \mathbb{R}$ which we denote as ∇f. Assuming we define $f : \mathbb{R}^n \to \mathbb{R}$, f will be defined on points in the the $\mathbb{R}^n$ space meaning that an input point p for f will have n dimensional components, and so we define the gradient of f at a particular point p as the vector

$$\nabla f(p) = \begin{bmatrix} \frac{\partial f}{\partial x_1}(p) \\ . \\ . \\ . \\ \frac{\partial f}{\partial x_n}(p) \end{bmatrix},$$

where $\frac{\partial f}{\partial x_i}(p)$ is the partial derivative of the scalar field f with respect the to the ith dimension of the $\mathbb{R}^n$ space with the value p plugged into it. $\frac{\partial f}{\partial x_i}(p)$ is a scalar because the partial derivative of f with respect to x_i will still be a scalar function meaning that once we plug in the components of p in the correct places in $\frac{\partial f}{\partial x_i}$ we again will be left with a scalar.

To unpack this idea further we physically think about the gradient as the rate of change of a scalar valued function across multiple dimensions essentially a muti-dimensional derivative for a function has multi-dimensional input values. Then once provided a n dimensional input p into the gradient, the resulting vector points in the direction of the scalar functions steepest ascent at point p. What we mean by the direction of steepest ascent is that the output value for $f : \mathbb{R}^n \to \mathbb{R}$ increases at the fastest rate in $\mathbb{R}$ when our input point p increases in the direction of our gradient at p.

0.2.1 Gradient example

Now we will provide an example of this, define $f : \mathbb{R}^2 \to \mathbb{R}$ as $\mathrm{f}(x, y) = 7 * \cos(.25 * x - \pi) + 7 * \cos(.25 * y - \pi) + 14$, The following graph is the visualization of the scalar field where the darker the area the greater the output value of $f(x, y)$.

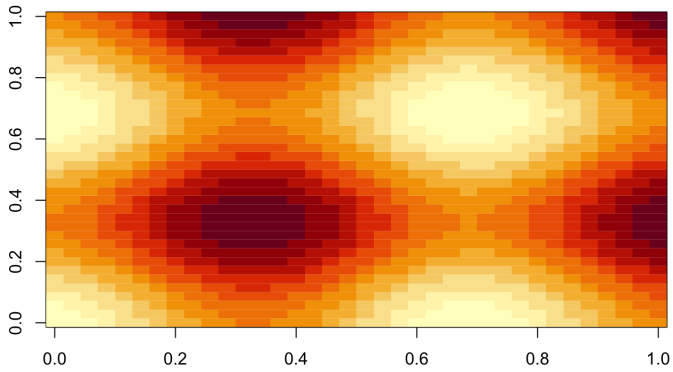

Figure 1: This is a scalar field of $\mathrm{f}(x, y) = 7*\cos(.25*x-\pi)+7*\cos(.25*y-\pi)+14$

now we define ∇f as

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x, y) \\ \frac{\partial f}{\partial y}(x, y) \end{bmatrix} = \begin{bmatrix} -7 * sin(.25 * x - \pi) * .25 \\ -7 * sin(.25 * y - \pi) * .25 \end{bmatrix}.$$

These output vectors are what we call a vector field and a vector field is simply a function on a space whose values are vectors. In this case the gradient of the function f is a vector field on the $\mathbb{R}^2$ space. We can visualize the gradient vector field below.

Now that we have an example of what it means to take the gradient of a scalar field in particular a real value scalar field, we can start thinking about what it means to find the divergence of a vector field. To start thinking about divergence we have to refer back to the gradient vector field we created earlier for $\mathrm{f}(x, y) = 7 * \cos(.25 * x - \pi) + 7 * \cos(.25 * y - \pi) + 14$. This vector field has arrows pointed in particular directions (vectors), and some of these vectors points towards a certain singular point, others point away from a singular point on all sides, and others point towards a point but are ultimately redirected in a different direction. All of these cases illustrates what we think of as flux for this particular scalar field or the amount of a theoretical substance passing through a particular point. If the net passage is inward, then the flux would be considered negative, and if the net passage is outward the flux would then be considered positive. Divergence is the notion of a rate of flux with respect to the volume

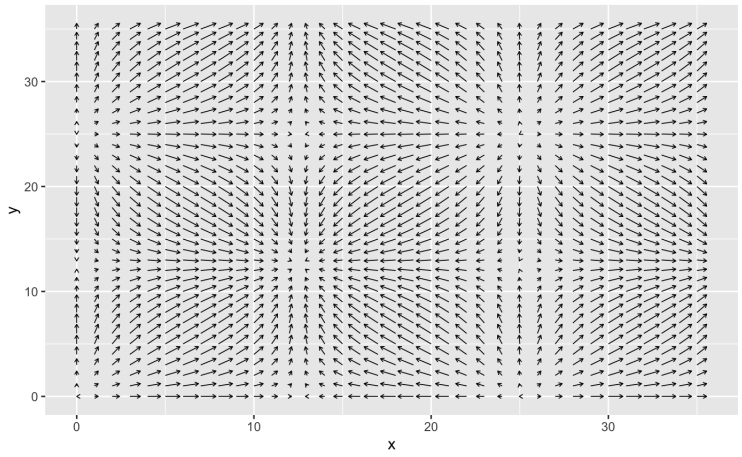

Figure 2: This is a vector field of the gradient of f$(x, y) = 7 * \cos(.25 * x - \pi) + 7 * \cos(.25 * y - \pi) + 14$

into or out of which the flux is moving. The divergence of a vector field $V(p)$ (also denoted $\nabla V(p)$) where p is an i dimensional point creates a scalar field defined in the following way:

$$\nabla V(p) = \sum_i \frac{\partial V_{x_i}}{\partial x_i}(p)$$

0.2.2 Divergence Example

Now we will provide an example of this, consider the vector field

$$g(x, y) = \begin{bmatrix} -7 * sin(.25 * x - \pi) * .25 \\ -7 * sin(.25 * y - \pi) * .25 \end{bmatrix}.$$

This is the gradient vector field we saw earlier and now we will use it as an example to calculate the divergence $\nabla g(x, y)$.

$$\nabla g(x, y) = -7 * .0625 * \cos(.25 * x - \pi) - 7 * .0625 * \cos(.25 * y - \pi)$$

Below is the graph of the scalar field this divergence of $g(x, y)$ creates.

We can compare the vector field of $g(x, y) = \nabla f$ to the scalar field we just created by observing those localities in the vector field where vectors on all sides are directed at a single point. These areas are the lightest colors on the scalar field because at these points the the divergence is considered negative because if we think about the word divergence it is a measure of the opposite of convergence. The white regions are regions of maximum convergence, the dark red regions are regions of maximum divergence or movement away.

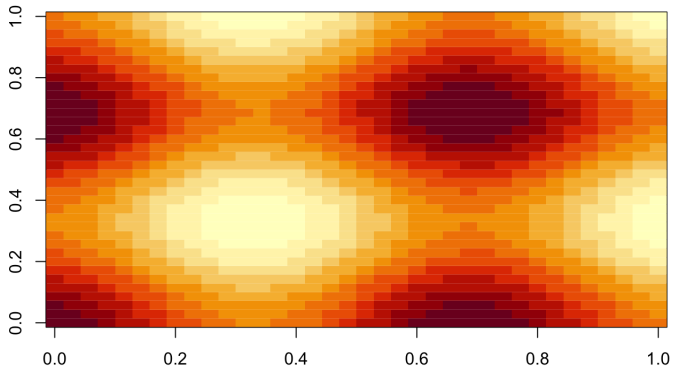

Figure 3: This is a scalar field of the divergence of the gradient of f$(x, y) = 7 * \cos(.25 * x - \pi) + 7 * \cos(.25 * y - \pi) + 14$

0.2.3 Intuiting the Laplacian

Now for the Laplacian, to understand the intuitive meaning behind the operator we must understand that the Laplacian of a Scalar valued function is the divergence of the gradient of that function. We explored earlier that the Gradient of a scalar valued function is essentially a multidimensional derivative for a scalar valued function with multidimensional input values. And for the divergence we perform a similar operation of partial differentiation as with the gradient, but instead, we sum the partial derivatives. Ultimately, we have found second partial derivatives (with respect to x and y) which we can then connect to the general idea of second order derivatives. While first order derivatives in single variable calculus are used to explore rates of change in a function, Second order derivatives in single variable calculus are used to explore the convexity and concavity of those functions. We understand this as positive second order derivatives implying a convex "u" region of the function and negative second order derivatives implying a concave "n" region of the function. We can use all of this to understand the meaning of the Laplacian which is in some sense the second derivative of a scalar field meaning that it tells us information about the concavity and convexity of the scalar field with respect to the multiple dimensions of the input values of the scalar field. Consider the xyz version of our original scalar field where the resulting scalar values of $f(x, y)$ are plotted on the z axis

In this xyz graph we can see that the 4 main concavities surround a convexity and these correspond to 4 dark red localities in the color grade scalar field from before surrounding a bright yellow locality. When we refer back to our divergence scalar field color grade graph we see that it is an inversion of our scalar field graph and this is because in single variable calculus we associate

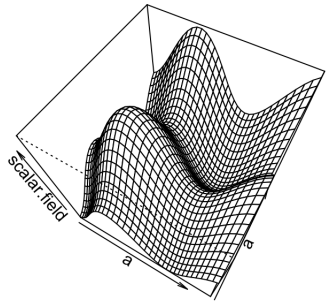

Figure 4: This is a 3d visualization of $\mathrm{f}(x, y) = 7 * \cos(.25 * x - \pi) + 7 * \cos(.25 * y - \pi) + 14$

concavities with negative second derivatives because the rate of change has to be steeply positive and then crosses 0 and becomes steeply negative, and we associate convexities with positive second derivatives because the rate of change starts as steeply negative then crosses 0 and becomes steeply positive. We see using this information that the divergence of the gradient of out original scalar field tells us information about the concavity and convexity of our scalar field and because of how we define the Laplace operator, this then means that the Laplacian of a scalar field tells us concavity and convexity information about that scalar field which we later ultimately generalize to topological information.

0.3 Graph laplacian and Heat Diffusion

We recall that the goal of this project is to analyze topological features of meshes and that an essential tool in doing this is the heat kernel signature (based on the laplacian) which tells us the curvature of the surface through which heat diffuses. Meshes are not completely smooth surfaces because in actuality they are triangulated approximations of smooth surfaces called simplicial complexes. This nature of the meshes causes them to resemble graphs especially in the way in which heat diffuses across their surfaces to reveal curvature features. This leads us to explore the concept of heat diffusion on a graph. Our end goal is for our exploration to lead us to the graph Laplacian which will tell us important information about graph concavity and convexity which will then allow us to compare meshes.

0.3.1 Diffusion In Connection To Curvature

We will derive the graph laplacian from the diffusion equation for graphs but before this we will derive the diffusion equation using physical understandings of diffusion itself. To derive the diffusion equation, we first invoke fick's law which says that a substance moves from regions of high concentration (for that substance) to regions of low concentration.

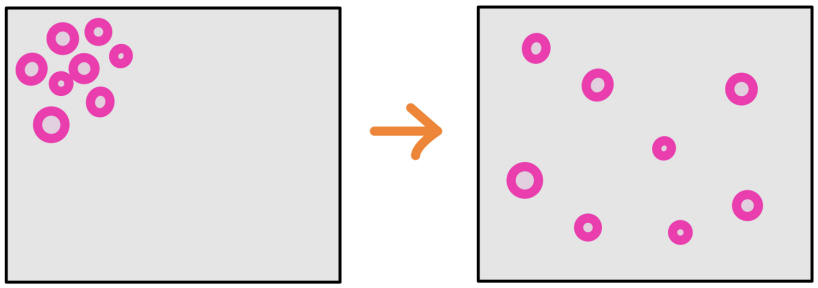

Figure 5: Fick's Law physically implies substance naturally moves from regions of high concentration to low concentration

We think about concentration of a substance as a function in terms of that substances distance spread through a surrounding solution and in terms of the time that has passed since its first infusion into the surrounding substance. We recall that flux is a mathematical concept describing rate of flow per unit area through a substance. We say that the diffusion flux of a substance is proportional to the change in concentration of a substance over change in distance that substance covers. In the context of more that one dimension this means that

$$\text{flux} = -D * \nabla \text{concentration}$$

where D is the diffusion constant and ∇concentration is the concentration gradient across all physical dimensions considered. If we were to only consider a single dimension then concentration gradient would be replace by the the partial derivative of the concentration function with respect to distance x.

We then invoke the conservation of mass equation to get

$$\frac{\partial \text{ concentration}}{\partial \text{t}} + \frac{\partial}{\partial x}\text{diffusion flux} = 0$$

we plug in from fick's first law to get

$$\frac{\partial \text{ concentration}}{\partial \text{t}} - D * \frac{\partial}{\partial x}(\frac{\partial}{\partial x}\text{concentration}) = 0$$

This then gives us the diffusion equation

$$\frac{\partial \text{ concentration}}{\partial \text{t}} = D * \frac{\partial^2}{\partial x^2}(\text{concentration})$$

we generalize this for multiple dimensions which then gives us

$$\frac{\partial \text{ concentration}}{\partial \text{t}} = D * \nabla^2(\text{concentration})$$

where the gradient squared becomes the laplcian and the laplacian tells us information about curvature. This last equation is what we call the diffusion equation and it takes on the the same form as the heat equation which is a partial differential equation that models the diffusion of heat through a surface. The only difference between these two partial differential equations is that the heat equation uses a thermal conductivity constant as opposed to a diffusion constant. Because of the similarities between the two equations, we mean to show that general diffusion is just as reasonable a phenomena to explore as heat diffusion when investigating the curvature of a surface. Now we we think about models of diffusion on graphs as opposed to smooth surfaces as our meshes while resembling at larger scales smooth surfaces are in actuality triangulated approximations of smooth surfaces that closely resemble graphs from close up.

0.3.2 Diffusion Through Graphs

In considering diffusion of a substance through the edges of a graph we start with the understanding that the diffusing substance starts with certain concentration levels at particular nodes in the graph. The edges between these nodes have particular diffusion rates between them so that after we begin the diffusion process, we can measure the change in concentration for each node by measuring the excess concentration of that node compared to surrounding nodes and observing the diffusion of that concentration by multiplying the excess by the edge rate of diffusion between whichever pair of nodes is under investigation. We focus on the excess because if two adjacent nodes have the same concentration levels then there wouldn't be a need for diffusion between the two since the

would already be at equilibrium not considering other nodes in the graph. In terms of calculations we say that if we have two nodes 1 and 2 and their respective starting concentrations are u_1 and u_2 then the the change in concentration over time from vertex 1 to vertex 2 is described by

$$\frac{du_1}{dt} = (\text{Edge Rate Of Diffusion}) * (u_2 - u_1)$$

Of course this would only partially describe the change of concentration for a particular node 1 because such a node would be connected to many other nodes meaning that many other edges would have to be considered. We take this into consideration by using summations and what is referred to as the adjacency matrix which tells us information about the connectivity of a node to the other nodes of a graph. We have the following differential equation to describe diffusion from node 1 to all adjacent nodes.

$$\frac{du_1}{dt} = \sum_{k=1}^{\text{Tot.nodes}} (\text{edge rate}) * \text{Adj}_{1k} * (u_k - u_1)$$

This then becomes

$$\frac{du_1}{dt} = \sum_{k=1}^{\text{Tot.nodes}} (\text{edge rate}) * \text{Adj}_{1k} * (u_k) - \sum_{k=1}^{\text{Tot.nodes}} (\text{edge rate}) * \text{Adj}_{1k} * (u_1)$$

$$= (\text{edge rate}) \sum_{k=1}^{\text{Tot.nodes}} * \text{Adj}_{1k} * (u_k) - (\text{edge rate}) * (u_1) \sum_{k=1}^{\text{Tot.nodes}} \text{Adj}_{1k}$$

We then say that

$$\sum_{k=1}^{\text{Tot.nodes}} \text{Adj}_{1k} = \text{ The number of edges connected to node 1 denoted } \deg_1$$

This then gives us

$$= (\text{edge rate}) \sum_{k=1}^{\text{Tot.nodes}} * \text{Adj}_{1k} * (u_k) - (\text{edge rate}) * (u_1) * (\deg_1)$$

and we then invoke the Kronecker delta to further expand $(\text{edge rate}) * (u_1) * (\deg_1)$. The Kronecker delta within the context of our example is defined as

$$\delta_{1k} = \begin{cases} 0 & k \neq 1 \\ 1 & k = 1 \end{cases}$$

We use this to create another summation so that

$$= (\text{edge rate}) \sum_{k=1}^{\text{Tot.nodes}} \text{Adj}_{1k} * (u_k) - (\text{edge rate}) * \sum_{k=1}^{\text{Tot.nodes}} (u_k) * \delta_{1k} * \deg_k$$

0.3.3 The Graph Laplacian Matrix

Now if we were to reconfigure these previous steps in terms of matrix multiplication we would say that $\sum_{k=1}^{\text{Tot.nodes}} \text{Adj}_{1k} * (u_k)$ becomes the adjacency matrix multiplied at the 1st row by a vector of node concentrations with dimension of the total number of nodes in the graph. We denote this operation with $[\text{Adj} * \vec{u}]_1$. For the other summation consider a degree matrix with information for each nodes degree along the diagonal and 0s at every other entry of the matrix. This matrix will then be multiplied only for the first row by $\vec{u}]_1$ which will isolate only the first node as we saw in previous iterations. This operation will be denoted by $[\text{Deg} * \vec{u}]_1$, and this leaves us with

$$\frac{du_1}{dt} = (\text{edge rate}) * [\text{Adj} * \vec{u}]_1 - (\text{edge rate}) * [\text{Deg} * \vec{u}]_1$$

and because its specialized for the first node but can be genrealized for all nodes, we say

$$\frac{d\vec{u}}{dt} = (\text{edge rate}) * [\text{Adj} * \vec{u}] - (\text{edge rate}) * [\text{Deg} * \vec{u}]$$

which gives us

$$\frac{du_1}{dt} = (\text{edge rate}) * (\text{Adj} - \text{Deg}) * \vec{u}$$

we see that this is essentially identical to our original formulation of the diffusion equation for smooth surfaces which indicates that the graph laplacian is basically the adjacency matrix minus the degree matrix, and this allows us to think about diffusion through a graph or eventually meshes as a way to explore curvature in those graphs or meshes.

0.3.4 Applications in Context: the HKS

The graph laplacian's application to our particular situation is that a function with respect to time the called heat kernel signature takes on the form

$$k_t(x, x) = \sum_{i=0}^{\infty} e^{-\lambda_i t} \phi_i(x) \phi_i(x)$$

where λ_i and ϕ_i are the ith eigenvalue and eigenfunction of the laplace beltrami operator which is the continuous version of the graph laplacian operator. The heat kernel signature essentially gives us a particular value for a point x at time t (of heat diffusion) on a surface that correlates to how much heat is transferred from x back to itself in said time. We need information about curvature, and this function gives us that information for multiple reasons. First how much heat remains at a point x on a surface tells us how connected this point is to other points on the surface on which heat is diffusing. Second, this gives us information about curvature because it tells us how much heat may return back to a point x

if diffusing to other points initially which requires we know about where point x is with respect to the rest of the surface again telling us information about the curvature at that point x. Lastly, we saw that the laplace operator tells us information about the curvature of a surface because it is directly linked with the second degree partial derivative which gives us information about concavity and convexity. This comes into play here when determining why we in fact need to apply this function to a surface we are observing to understand its curvature. Ultimately after applying the HKS to the objects under investigation in this thesis, we will then be able to think about what topological features exist at certain points on the surface once we apply the concept of simplicial persistence homology and filtrations which we will explore in the next section.

0.4 Simplicial homology

0.4.1 Simplicial Complexes

We Define a **simiplicial complex** C as a collection of subsets of a set C_0 satisfying the conditions that all Subsets are Non-empty and that If c_i is a subset in the collection of C, all non-empty subsets of c_i are also individual elements in the simiplicial complex C. Subsets c_i with p+1 elements from C_0 are called p-simplices. Consider the following example. Let $C_0 = \{a, b, c, d, e\}$ be a set. Now define the simplicial complex

$$C = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{c, d\}, \{a, b, c\}\}$$

The graphical representation of a simplicial complex is called its **geometric realization** where elements of C_0 are vertices of the graph and non-zero simplices generate physical relationships between the vertices. 1 simplices denote edges between vertices. 2 simplices denote a filled in triangle of 3 vertex interaction.

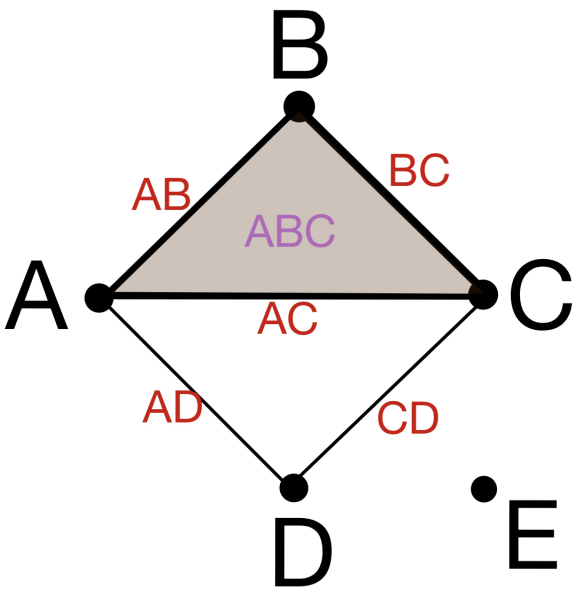

Figure 6: This simplicial complex has 1 2-simplex, 5 1-simplices, and 5 0-simplices

In the Simplicial complex C we say that the 0 simplicies are $\{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$, then the 1 simplicies are $\{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{c, d\}\}$, and lastly, the only 2 simplex is $\{a, b, c\}$. Simplicial complexes can be much more complex however within the context of this project we will only be focusing on at most a collection of 2 simplices that ultimately approximate a smooth surface. These collection

of 2 simplices will make up what we refer to as a mesh or a triangulated approximation of a smooth surface. Putting this into this perspective will allow us to observe topological information within the context of these graph like structures as opposed have to refer to continuous topological methods to find information about approximated surfaces.

0.4.2 Topological Connection

In order to start analyzing the topological characteristics of these simplicial complexes, we have to find some aspect of these complexes to connect to topological characteristics as whole, which leads us to boundaries of the simplices.

To start:

1. The boundary of a 0 simplex is 0 since points have no boundaries
2. The boundary of a 1 simplex are the two endpoints. In the example above, this definition means that the boundary of $\{a, b\}$ is a and b, the boundary of $, \{a, c\}$, is a and c, the boundary of $\{a, d\}$ is a and d, the boundary of $\{b, c\}$, is b and c, and the boundary of $\{c, d\}$ is c and d.
3. The boundary of a 2 simplex is the edge of the area space. The outline of the triangle as opposed to the filling of the triangle. This applies to the simplicial complex listed above in that the boundary of

$\mathbf{C}_p(C)$ is the vector space on $\mathbb{F}_2$ where the basis vectors are those p simplices of the simplicial complex

For Instance the basis for the $\mathbf{C}_1(C)$ is

$$\{\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{c, d\}\}$$

Consider Maps between vector spaces

$$d_p : \mathbf{C}_p(C) \to \mathbf{C}_{p-1}(C)$$

where some p simplex is sent to the sum of those p-1 simplices that are subsets of the p simplex.

Essentially a simplex is being sent to its boundary. When p = 0 d_p is the 0 map.

Now for an example, consider within the context of our simplicial complex above

$$d_1 : \mathbf{C}_1(C) \to \mathbf{C}_0(C)$$

as

$$d_1 = \begin{bmatrix} 1, 1, 1, 0, 0 \\ 1, 0, 0, 1, 0 \\ 0, 1, 0, 1, 1 \\ 0, 0, 1, 0, 1 \\ 0, 0, 0, 0, 0 \end{bmatrix}$$

The reason d_1 takes this form is that each row represents a particular 0 simplex that is either present or absent in the boundary of a particular 1 simplex, and each column represents a 1 simplex in the simplicial complex C. The rows are in alphabetical order going down, and The order of these columns is also alphabetical. The boundary relationships between particular 0 simplicies and 1 simplicies is signified by either a 1 (for present) or a 0 (for absent) being placed in the intersection of the particular row and column. For example there is 1 at the intersection of the first row and first column which means that 0 simplex a is part of the boundary of 1 simplex ab.

and also consider that

$$d_2 : \mathbf{C}_2(C) \to \mathbf{C}_1(C)$$

Is represented as

$$d_2 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

where each row represents a 1 simplex that may or may not be present in the boundary of a 2 simplex. again the rows are listed in alphabetical order in the same way that the columns for d_1 are listed in alphabetical order. We see that the intersection of the first row and first column is 1 because the 1 simplex of ab is present in the boundary of the only 2 simplex in this simplicial complex which is the solid triangle abc.

Because we will only be using at most 2 simplicies in this project since it is through triangular approximation that is how our meshes will replicate hollow surfaces which we will then analyze. If we were to consider $p > 2$ we would follow a similar pattern as we did for d_1 and d_2 where the rows represent the $p-1$ simplicies that may or may not be present in the boundary of each of the p simplicies in the simplicial complex, and the columns represent the the p simplicies.

We said that d_p sends a p-simplex in a simplicial complex to the sum of the $p-1$ simplicies that make up its boundary. d_1 sends edges to points that define edges, and d_2 sends solid triangles to the sum of edges that define those triangles. We are beginning to build the tools necessary for a topological understanding of simplicial complexes. To add to our toolkit, we must conceptualize holes and connected components, and to do this we must take advantage of the linear algebraic principles that the matrices d_p abide by. When we conceive of a hole we should first think about 2 simplicies as disks and their boundaries being a circle or a sum of the individual 1 simplicies around the disk. Those circles that are not the boundaries of some disk are hollow and therefor holes. We can identify these sum of 1 simplices that are cycles by applying the boundary operator to them and observing them go to 0. Cycles under the boundary operator d_1 will

go to 0 because there isn't a single 0 simplex the cycle can be said to come from and end at. In other words, the number of holes in simplicial complex C is

$$\dim(\text{kernel of } d_1) - \dim(\text{image of } d_2)$$

We can extend this idea to k-dimensional holes recalling that we just observed 1-dimensional holes or cycles for our simplicial complex C.When K is 0 we are counting the number of connected components. The number that we get from all of this is called the kth betti number and it is a essential topological characteristic that we can use to learn more about the structures of the meshes in this project. We have that:

$$\text{betti}_k = \dim(\text{kernel of } d_k) - \dim(\text{image of } d_{k+1})$$

To explore the more technical aspects of this concept we can explore how each of the components of the betti number equation for a simplicial complex are found using our example simplicial complex C and the k value 1. To find the dimension of the kernel of d_1 we invoke the rank nullity formula which is that given a linear transformation d_K

$$\text{Rank}\{d_K\} + \text{Nullity}\{d_K\} = \dim\{\text{domain}\{d_K\}\}$$

recall that we said the basis of $C_1(C)$ are the 1 simplices of which there are 5 which means the dimension of the domain of d_1 is 5. Further we intend to use this theorem to find the dimension of the kernel of d_1 which is the same as the nullity of d_1. We can do this by finding the Rank of d_1, and to do that, we put d_1 into row echlon form (within the context of $\mathbb{F}_2$):

$$\begin{bmatrix} 1,1,1,0,0 \\ 0,1,1,1,0 \\ 0,0,1,0,1 \\ 0,0,0,0,0 \\ 0,0,0,0,0 \end{bmatrix}$$

After this we count the number of non-zero rows which is 3 and that gives us the rank of d_1. We use this to find the nullity which is 2 and again the nullity is the dimension of the kernel of d_1. Now we need to find the dimension of the image of d_2. We said that the domain which is $C_2(C)$ has a basis of the only 2 simplex meaning that the dimension of the domain is 1. We need the rank and for the dimension of the kernel we have 0 because there are no elements in the $C_2(C)$ domain that are sent to 0 under the boundary operator. We see that the rank is 1 meaning the dimension of the image of d_2 is 1. This then tells us that $\text{betti}_1 = 2 - 1 = 1$ This means that there is only a single hole in our simplicial complex. We use this basic topological understanding of simplicial complex via linear algebra in order to explore the structures of our triangulated meshes using persistence diagrams where at each level set we topologically explore a simplicicial complex.

0.5 Persistence Diagrams

While the Graph Laplacian is in fact what is used to map the curvature using a black and white gradient on the meshes in this project, on our end, we must explore persistence diagrams in order to much more closely investigate the curvature of the simplicial complexes that make up these meshes in order to find similar topological characteristics between these meshes so that we may be able to recognize them in different positions. A persistence diagram is an applied topological tool that describes characteristics of a geometric object using the succession of sub-level (explain further) sets with the passage of time. On the persistence diagram the persistence of particular topological characteristics of a geometric are graphed. The x-axis of a persistence diagram is labeled birth while the y-axis is labeled death, and these come from the same measurement of time on a geometric object. A persistence diagram can be generated from many different types of geometric objects ranging from real valued functions to a set of n-dimensional disks.

0.5.1 Sub-Level Sets and Persistence Diagram Examples

Pictured below are successive sublevel sets of a ring of balls from an overhead two dimensional perspective.

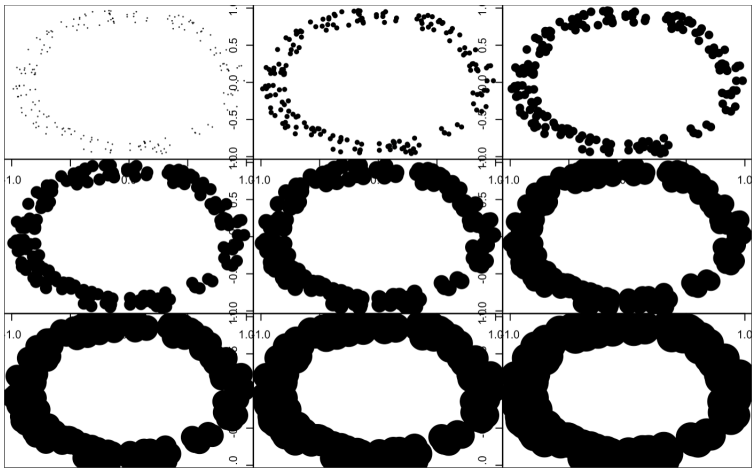

Figure 7: These are 9 different filtrations of a collection of balls with separation based on sublevel sets along the z axis perpendicular to the page

As we move left to right from the top left graph with 200 connected components which are originally just dots, we see that the respective radii of those dots expands resulting in overlaps of the connected components resulting in fewer of them. A persistence diagram represents the convergence of connected components by plotting a point on an xy plane where x is the birth time of a

connected component and y is the death time of a connected component. Each instance that 2 connected components converge to become a single one, one of the two dies and and this is the death time used for the y coordinate.

If we look closer we see that sometimes there is open space that is created by connected components on all sides which is essentially a hole. As the radii of the connected components grow, these holes will fill in because of the excess radius, and their existence will be plotted on the persistence diagram by a point where the birth is the instance at which the bubble of space was originally formed and the y coordinate will be the instance at which the space is completely filled in. So far we have explored 0 dimensional holes (connected components) and 1 dimensional holes (literal holes). We might consider higher dimensions, however, this would be past the scope of this project because the simplicial complexes we are investigating only go up to a dimension of 2 which results only in the observation of at most 1 dimensional holes.

The separate instances that are looked at above through the process of these expanding spheres are called filtrations, and this is because aspects of the topological objects along with features that appear with those aspects are filtered out from the larger geometric whole through the application of particular times or function values.For our purposes we will define filtrations of geometric objects on sublevel sets of those objects.

We could also consider another example where instead of radius expanding discs as objects for which we consider persistence homology, instead we consider the convergence of minima of a polynomial as we raise a horizontal line through the range of that function as the time inducing element on which we explore birth and death times. This idea is pictured below.

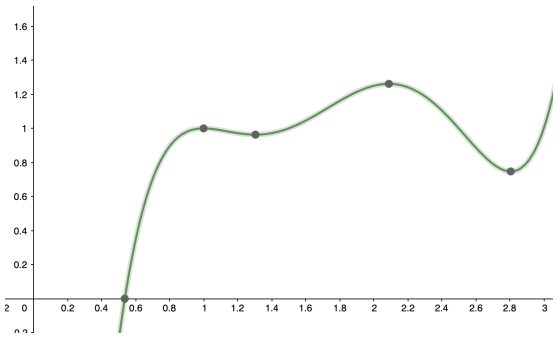

Figure 8: Credit of Desmos Graphing Calculator, A polynomial with extema and zero labeled with black dots

Above we see the original polynomial we will be constructing a persistence diagram off of. When dealing with polynomials, the topological features to be explored are local and global extrema since the function is already continuous.

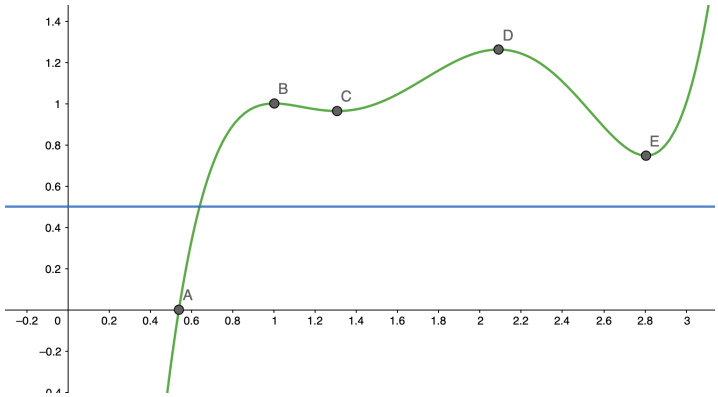

Figure 9: Credit of Desmos Graphing Calculator, A subleveling axis (blue) to create different filtrations on the polynomial as it rises along the y axis

A time t sub level set of real valued function f is defined as the set $\{x \in Domain(f) | f(x) \leq t\}$ these sub level sets may be compose of separate connected components depending of the value for t. These would be continuous intervals between 2 numbers.

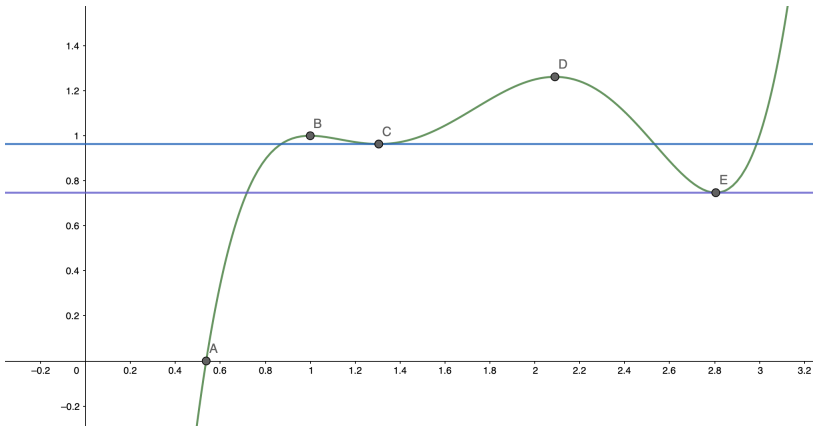

Figure 10: Credit of Desmos Graphing Calculator, A subleveling axis (blue) creates a birth date (y value) for a connected component on the polynomial as it rises along the y axis and hits a local mininma

Birth time of connected component x is the y value at which the horizontal line intersects the local minimum from which our component expands. Here we see that $\text{Birth}(E) = .74712$ and $\text{Birth}(C) = 0.96325$.

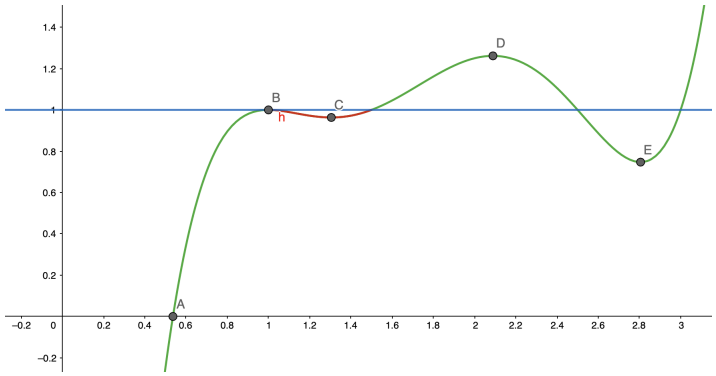

Figure 11: Credit of Desmos Graphing Calculator, A subleveling axis (blue) creates a death date (y value) for a connected component on the polynomial as it rises along the y axis and connects a convexity with a deeper convexity

A Death time for connected component x is the y value at which x expands to reach a local maxima and the local (or global) minima on the opposite side of that local maxima is deeper (or older) than x. We see that Death(C) = 1. We plot this on a persistence diagram using the birth and death times for point C and we ultimately get the persistence diagram below.

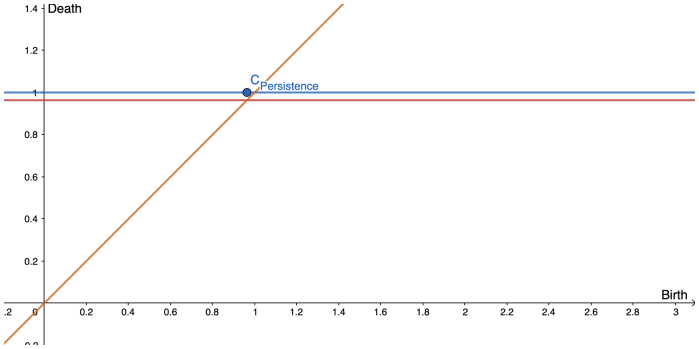

Figure 12: Credit of Desmos Graphing Calculator, A persistence diagram graphs the persistence of a topological feature by graphing (birth date, death date) of a feature.

0.5.2 Specialization

Ultimately these diagrams can graph many different types of topological information depending on what is the geometric object of focus. Our priority in this project is observing the persistence of 1-dimensional holes and connected components in simplicial complexes in our meshes.

Below we have an example of a simplicial complex from one of our meshes and inside each 2 simplex we have a number associated to the time at which that 2 simplex is contained in a time sublevel set dependent on the z axis orthogonal to the page. The first 3 complexes from left to right are for times 0, 1, and 2 respectively, and the bottom 3 are for times 3, 4 , and 5. We see that at time 1 we have two separate blue highlighted regions which represent 2 connected components. At time 2, we have pink highlighted regions connecting those two blue regions to create a single cycle and a single connected component. This creates our first black point on our persistence diagram. Then on the second row we have another connected component inside of the hole we've created which gives us another black dot, and at the 4 instant, the hole we created is reduced in size as the new connected component merges with the larger cycle to create a single connected component. This instant creates as black dot on the persistence diagram. At the fifth instant, the hole is filled in, and we are left with a single connected component. This creates a light blue dot on the persistence diagram. The final connected component lasts for eternity as this fact is also graphed on the persistence diagram as the highest black dot.

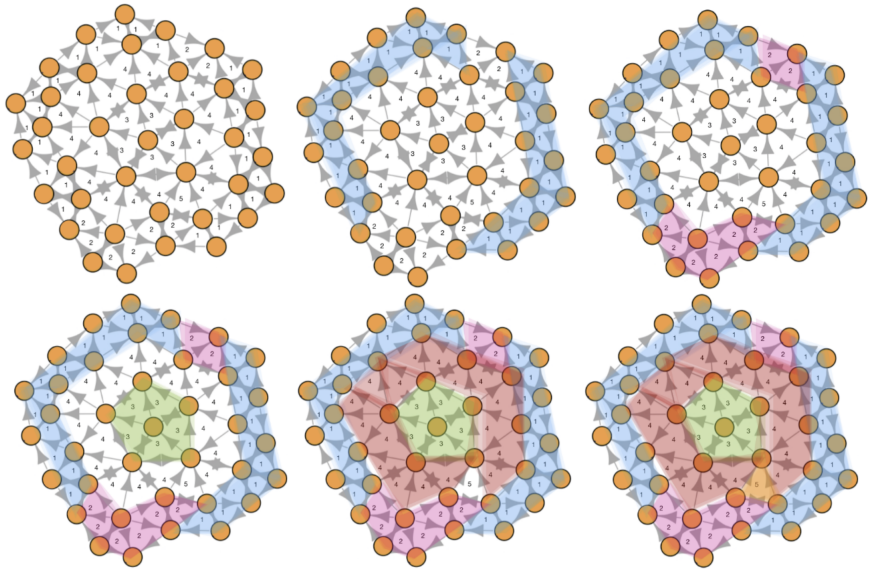

Figure 13: We have 6 different filtrations of a simplicial complex that gives us information about curvature, and 3 dimensional structures. The filtrations are based on numberings of the 2-simplicies which are solid triangles

Now we can take a loo at the final persistence diagram we are left with after our analysis where connected components persistence information is graphed by black dots and 1 dimensional hole's persistence information is graphed by plotting blue dots at intersection of birth and death times. Additionally we have diagonal line $y = x$ axis in order to convey the idea that it is impossible for the death time of a topological characteristic to be earlier or lower numeric value that the birth time of that topological characteristic. Otherwise the that characteristic would have never existed and thus have no reason to be plotted to begin with on the diagram.

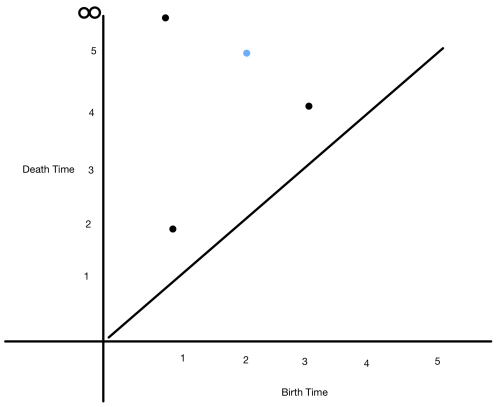

Figure 14: We plot persistence information about the features of our simplicial complex we found through the passage of the different filtrations. We plot holes and connected components for this project since we only go up to 2-simplices.

If we zoom out to see the larger picture, we see that the mesh below corresponds with the persistence diagram to the right displaying the survival and deaths of particular connected components within the larger mesh.

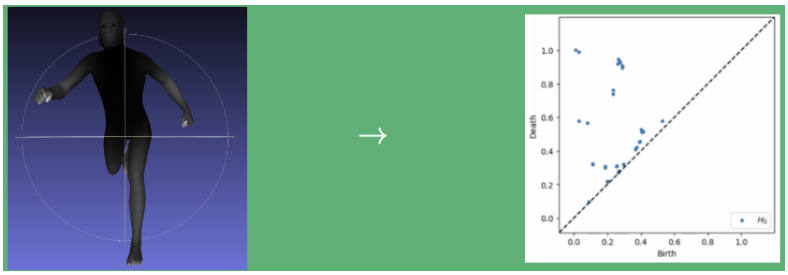

Figure 15: We can get particular persistence diagrams from entire body meshes once we apply the heat kernel signature to the mesh itself and use sub-leveling filtration. (Source: Meshlab and Rstudio)

The reason why we need a function (the HKS) signifying curvature to be applied to these meshes is that when thinking about categorizing the meshes by body type, we have to consider the fact that the different positions require a different set of points in 3 dimensional space when being visualized in meshlab. This means that in order to discover common features across different meshes, the positions within 3 space will not give us enough information. Instead its more helpful to be given information about the curvature of a chest, a finger or an arm because this information will remain consistent regardless of the position of the mesh, and this is because the HKS has the property that it is invariant under isometries or non-distorting transformations with respect to the simplicial complexes. We compare the topological features or the features related to curvature by analyzing the persistence diagrams we are able to generate from these meshes.

0.6 Kernel Trick

0.6.1 Motivation

The kernel trick is a tool that's used in machine learning to analyze patterns in data such as clusters, correlations, and classifications. The Motivation for the kernel trick is that many times the relationships that are looked for are related in higher dimensions that can not be easily seen on the euclidean plane. The kernel trick as a machine learning concept interrogates the closeness of a data point of focus to other points already present in the training data.

0.6.2 General Idea

However, because those relationships being explored are non-linear, to save time the data is projected into higher dimensional space where distance is once again explored, but however this time, distance can be found using the inner product of the higher dimensional space that is projected into, but the key to the kernel trick is that both of these steps, first the projection and then the inner product of the two points within that projection space, can be written using a kernel function or a function on just the input data space. In other words, one can predict higher dimensional output values for input values of test data based on proximity to input training data values without having to explicitly project into higher dimensional space. Another essential aspect to this idea of the kernel trick is the idea of the inner product which allows us to investigate the angle between vectors in a particular space. This is because numerically, the inner product ranges from -1 to 1 where -1 indicates that the 2 vectors being analyzed point in opposite directions, but if 1, the two vectors are pointing in the same direction. If the inner product is 0 this means that the two vectors are orthogonal which ultimately means that they are as unrelated to each other as possible. According to mercer's theorem, any positive definite function is a kernel function. In other words,

0.6.3 Example Situations and Kernels

Consider a situation where the input data values are in $\mathbb{R}^2$ space and the kernel is meant to project into the $\mathbb{R}^3$ space. One kernel function that projects into the $\mathbb{R}^3$ space and then finds the inner product between elements of the 3 space is given by the equation

$$K(x, y) = x \cdot y + ||x||^2||y||^2$$

and this is where the projected third dimensional component of the data points is calculated by adding the squares of the first and second dimensional components together. We see that this equation is in fact the same as the inner product of two vectors in $\mathbb{R}^3$ when we expand the inner product of two vector in $\mathbb{R}^3$ a few level taking into consideration the method through which the third dimensional component is calculated for input data:

Let

$$(x_1, x_2, x_3), (y_1, y_2, y_3) \in \mathbb{R}^3$$

Then

$$(x_1, x_2, x_3) \cdot (y_1, y_2, y_3) = x_1 y_1 + x_2 y_2 + x_3 y_3$$

we apply the projection calculation

$$= x_1 y_1 + x_2 y_2 + (x_1^2 + x_2^2)(y_1^2 + y_2^2)$$

and we recognize the inner products in the input space

$$= x_1 y_1 + x_2 y_2 + (x \cdot x)(y \cdot y)$$

From linear algebra, we see that this is the same thing as

$$= x_1 y_1 + x_2 y_2 + ||x||^2 ||y||^2$$

and this was the kernel function that we began with so through this particular example we see that the kernel function is actually capable of projecting data into higher dimensions and finding the inner product between those projections only using the input data to start with. The kernel trick is used to explore patterns particularly this example can be used for the classification of data that is distributed in a radial fashion in the $\mathbb{R}^2$ space.

We will express this graphically to see how this particular example may be used for binary classification. Consider these set of points in the $\mathbb{R}^2$ space.

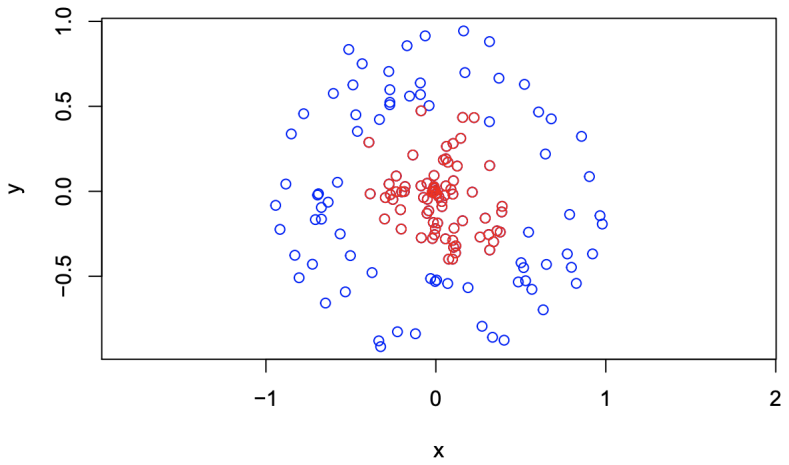

Figure 16: When considering machine learning uses for the kernel trick one prominent example is when we have to separate data that scatters in a similar to the above

When in action a visualization of the projection may not be available because of higher computational intensity however, we can think of the projection behind the scenes resembling the following graphic in 3 dimensions.

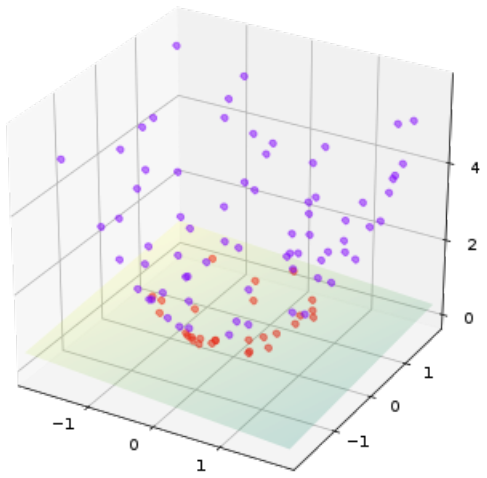

Figure 17: Credit of Wikipedia, A part of the kernel trick is the projection of data into higher dimensions in order to create easier computational analysis of the data and its trends. In this case 2-d data is projected into 3 dimensions.

A classifier within the larger context of machine learning would be applied to this and the central points would be separated from the peripheral points to create predictions about where a point is red or blue. Additionally, we see that there is already the presence of curvature once the kernel trick's projection goes into action which is really important to the rest of our project because it means that we may be able to apply the projection principles of the kernel trick to larger data sets to explore the simplicial complexes of the meshes we explored earlier.

0.6.4 Radial Basis Kernel

The radial basis kernel function is a kernel with a feature map that goes into an infinite dimensional feature space $\mathbb{R}^\infty$. The ultimate map takes on the form of this function is $K(x, y) = e^{-\gamma||x-y||^2}$, where x and y are two points in the data and γ is a parameter that can be adjusted to influence the fitting of the decision boundary for the data. To be more specific, the radial basis function generates a decision boundary using the layering of multiple Gaussian curves as if creating a composition of functions. The parameter gamma tells us information about the standard deviations and spread of these Gaussian curves which in turn tell us

information about how finely tuned to the training data the ultimate decision boundary is.

0.6.5 Application of Kernels

Now that we have defined the concept of kernels we can begin to consider applications of the concept in context of this project. We can compare persistence diagrams to each other to find closeness of certain features on meshes to each other using the persistence scale space kernel (PSSK) in order to see how similar separate meshes are to each other. The PSSK is actually very similar to the RBF which means that it is easy to use for our purposes. Ultimately our project intends to generate machine learners that could classify test object data using a learner trained by other object training data.

0.7 Support Vector Machines

Support Vector Machines is a machine learning technique that involves the finding a linear boundary that divides numeric data along two classes. For instance in the picture below we see that 2 dimensional data points take on either the color red or green and applying a linear boundary to this data will attempt to find the optimized linear boundary between these two classes within the data.

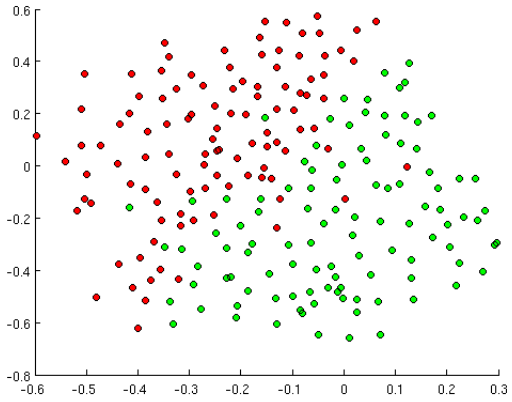

Figure 18: 2 dimensional Data of two classes credit of Stanford open classroom

Below you can see an example of a linear boundary layed over the data in order to separate the red from the green.

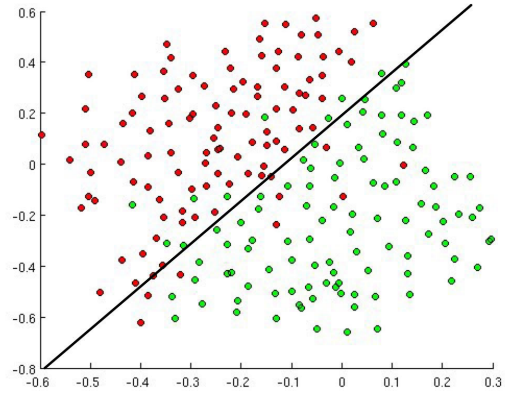

Figure 19: 2 dimensional Data of two classes credit of Stanford open classroom

The process of optimizing where this decision boundary is is at its most

basic level determining what line is furthest from from the nearest points on both sides. In other words we must find the line whose orthoginal distance is greatest on both sides from surrounding points. At the same time we must consider parameters related to when the classes are heterogeneous and these take the form of mis-classification cost which has to do with how big of a problem with the model each mis-classification is considered once the decision boundary is drawn.

0.7.1 Non-linearly separable data sets

Sometimes data is not distributed in such a way that it can be divided for the most part along a linear boundary. We can recall figure 16 as an example of this. We can apply support vector machine process in the feature space just as we did when we were able to remain in 2-space. Instead of a line though, the decision boundary's dimension will be dependent on the feature space's dimension. For instance a decision boundary in 3-space will be a plane. Ultimately, however, when we see the projection of this decision boundary into the original two space we see a non-linear decision boundary created and optimized using the same methods when not utilizing a feature space. Below is an example of how we can get such a non-linear boundary.

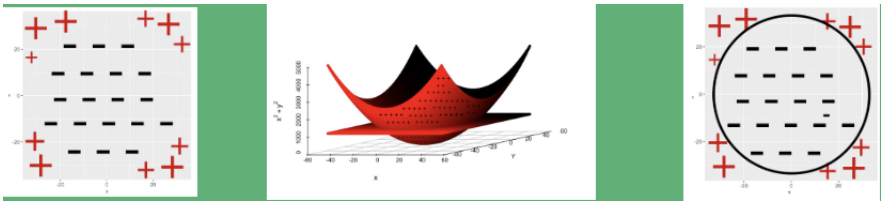

Figure 20: Credit of r-studio

Many times when using a kernel called the radial basis function, an additional parameter must be explored in terms of how closely the decision boundary fits training data with respect to what dimension the feature space is, and this is because generally the rbf can map into infinite dimensional feature spaces.

0.7.2 After the Boundary Is constructed

Once we construct a decision boundary using the training data, for additional points not included in the training data that we would like to classify, we can use the decision boundary to classify these. Ultimatly all of these steps can be reduced into a single step using the kernel trick which classifies a point based on its inner product with the points on the training set in the feature space for our purposes well be using the pssk to ultimately classify meshes according to the 15 body types of classes.

0.8 Results

Ultimately, after completing the procedures, Huber, Bauar, and Kwitt found in their 2014 Paper, *A Stable Multi-Scale Kernel for Topological Machine Learning* that the success rate correctly classifying synthetic meshes with respect to the 15 body types or classes was 99.3 ± 0.9, while for the real data meshes, the rate of success was around 62.7 ± 4.6. While the second result may seem low its still a very good rate of success because of the fact that this is a 15 class classification problem instead of just a 2-class problem. The reason it is lower than the synthetic data has to do with the fact that real data is harder to recognize isometric transformations with than the synthetic data because in reality the surface changes a bit more.

Bibliography

[Baas et al., 2020] Baas, N., Carlsson, G., Quick, G., Szymik, M., and Thaule, M. (2020). *Topological Data Analysis: The Abel Symposium 2018*. Abel Symposia. Springer International Publishing.

[Bauer and Lesnick, 2020] Bauer, U. and Lesnick, M. (2020). Persistence diagrams as diagrams: A categorification of the stability theorem. In Baas, N. A., Carlsson, G. E., Quick, G., Szymik, M., and Thaule, M., editors, *Topological Data Analysis*, pages 67–96, Cham. Springer International Publishing.

[Berry et al., 2018] Berry, E., Chen, Y.-C., Cisewski-Kehe, J., and Fasy, B. T. (2018). Functional summaries of persistence diagrams.

[Cohen-Steiner and Edelsbrunner, 2005] Cohen-Steiner, D. and Edelsbrunner, H. (2005). Inequalities for the curvature of curves and surfaces. volume 7, pages 272–277.

[Marchese, 2017] Marchese, A. (2017). Data analysis methods using persistence diagrams.

[Pickup et al., 2014] Pickup, D., Sun, X., Rosin, P. L., Martin, R. R., Cheng, Z., Lian, Z., Aono, M., Ben Hamza, A., Bronstein, A., Bronstein, M., Bu, S., Castellani, U., Cheng, S., Garro, V., Giachetti, A., Godil, A., Han, J., Johan, H., Lai, L., Li, B., Li, C., Li, H., Litman, R., Liu, X., Liu, Z., Lu, Y., Tatsuma, A., and Ye, J. (2014). SHREC'14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, EG 3DOR'14. Eurographics Association.

[Reininghaus et al., 2014] Reininghaus, J., Huber, S., Bauer, U., and Kwitt, R. (2014). A stable multi-scale kernel for topological machine learning.

[Sun et al., 2009] Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum*.