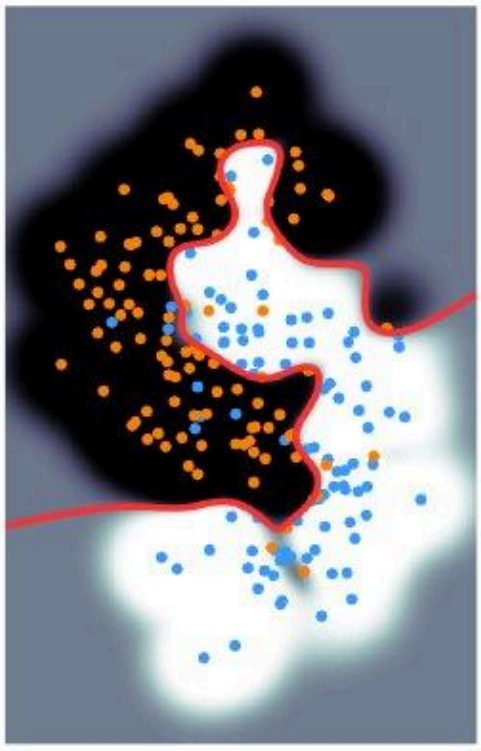
A Topological Regularizer for Classifiers via Persistent Homology

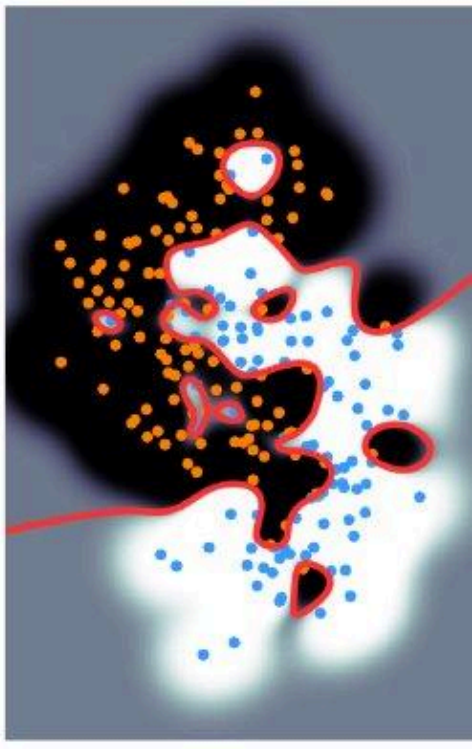
Authors:Chao Chen, Xiuyan Ni, Qinxun bai, Yusu Wang

Code: https://github.com/tanmay2233/Topology_SRIP.git

INTRODUCTION:

In the paper, they have proposed the idea of reducing the complexity of invariant topological structures. In this case, they have proposed a way of reducing the number of connected components of a machine learning model as a way of reducing the overfitting of data, so that the model may perform better against unseen data.
In the below image, we can clearly see that the second image is overfitting by taking every point of class-0 under it's boundary by the help of hyper-parameter tuning of the SVM classifier. Thus, we are navigating ways of reducing the number of connected components as shown by the first image below.
This is done by utilising the robustness of topological structures, and introducing topological penalty as regulariser as discussed in depth in the following section.

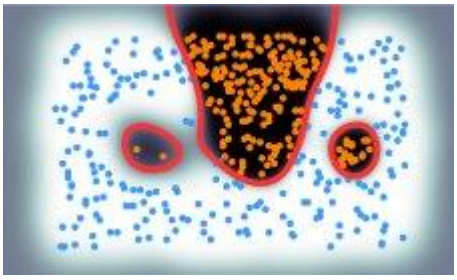




Brief Overview of Methodology:

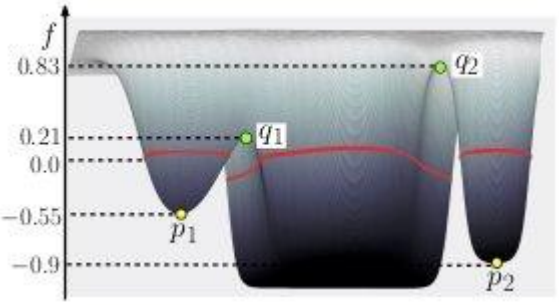
Following are the key terminologies related to the methodology

Robustness:



- Robustness is used to quantify how significant a connected component is.
- So, robustness is defined as the minimum amount of disturbances to be introduced in a connected component in order to remove it off from the boundary of the classifier.
- For example, in the figure above, the left loop is less robust as it has been formed due to lesser number of points as compared to the right loop, which requires a larger relatively amount of noise to be introduced in order to detach it from the classification boundary & merge with some other connected component.

PERSISTENCE PAIRINGS:



In the persistence diagram, for each feature we make a pairing of the form (p, q) where the point p is the birth point and q is the death point or a point where it merges with another feature. Generally, the birth point is the minima and the death point is a saddle point.

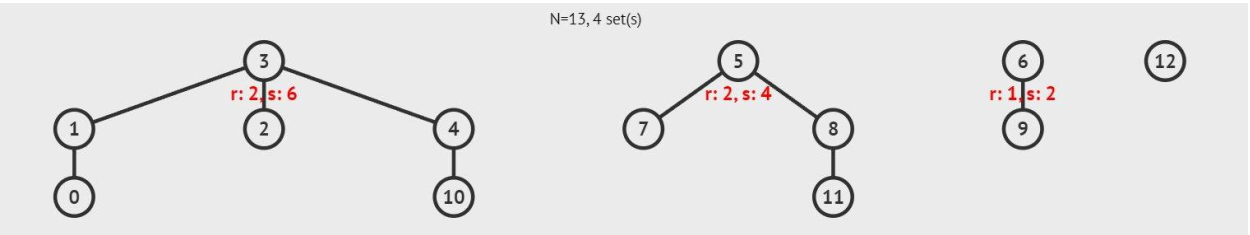
For example, in the above figure, p1 is the birth point of a feature and q1 is its death point.

Detailed Algorithm

To efficiently compute the critical pairings representing the birth and death of features, we define the set **Π S_f** such that it is the collection of all the critical pairs of the form **(p, q)** where p is the **birth** point and q is the **death** point of the feature and **f(p) <= 0 & f(q) >= 0**.

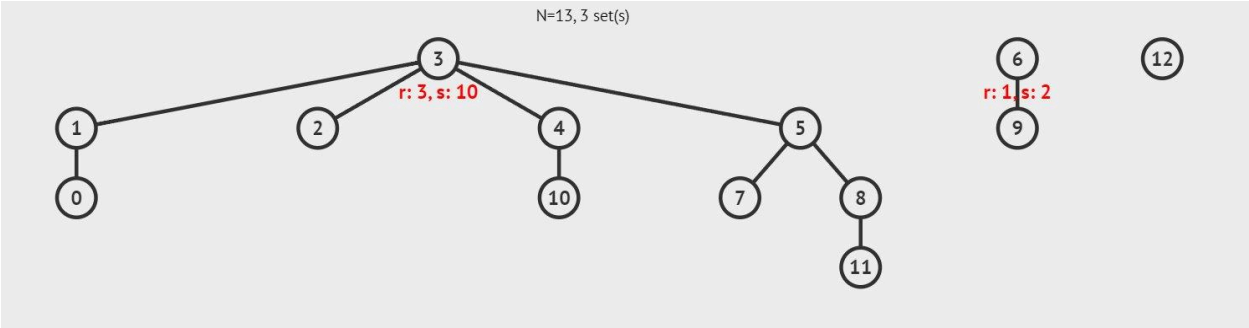
This set has a collection of the critical pairs of all those components which persist on the boundary of the classifier. This would help in calculating the robustness of each component.

- Each point is considered as a vertex and the vertices are arranged in increasing order of their function values to form sublevel sets as we sweep through the entire domain.
- A spanning forest is maintained in which every tree represents a connected component represented by the global minimum value in that tree.
- Whenever a vertex v_i is added and it leads to the merging of 2 trees T1 & T2 (minimas p1 & p2 & p1 < p2), the resulting tree is represented by p1.
- The pair (p2, v_i) is added to the set as this component was born at p2 and merged into another component at point v_i.
- Same process is repeated for -f to capture the components in the opposite direction of function values.
- Take a union of the pairs obtained from f and -f.



Initial tree forests

For example, if these are the trees formed after **i** steps and due the introduction of the **(i+1)th** vertex, the first 2 trees merge the merged tree would look as follows:



Union Find of 5 and 3

This figure shows the resultant tree after the merging of the first 2 trees. This actually shows the merging of 2 connected components features. After this step, the corresponding critical points' pair would be added into our set of critical pairs.

Topological Penalty & Gradient Calculation

- Since the domain of the function could be such that closed form solutions of critical points might not exist, making the gradient descent optimization difficult.
- To deal with this problem, the classification function is approximated to a piecewise linear function to make it differentiable everywhere.
- After this, the loss function to optimize is as follows:

  $$L(f, \mathcal{D}) = \sum_{(x,t)\in\mathcal{D}} \ell(f(x, w), t) + \lambda L_{\mathcal{T}}(f(\cdot, w))$$

  Where the first term is the per data loss function like quadratic, hinge loss, etc. and the second term is the regularization term composed of the regularization parameter and the Topological penalty function.
- Topological penalty here is the sum of squares of each component's robustness value.
- In each iteration, the gradient descent algorithm is further optimizing the parameters of the model due to the added Topological penalty to avoid overfitting.

RESULTS GIVEN IN THE PAPER:

Table 1: The mean error rate of different methods.

Synthetic							
	KNN	LG	SVM	EE	DGR	KLR	TopoReg
Blob-2 (500,5)	7.61	8.20	7.61	8.41	7.41	7.80	**7.20**
Moons (500,2)	20.62	20.00	19.80	19.00	19.01	18.83	**18.63**
Moons (1000,2,Noise 0%)	19.30	19.59	19.89	17.90	19.20	17.80	**17.60**
Moons (1000,2,Noise 5%)	21.60	19.29	19.59	22.00	22.30	**19.00**	**19.00**
Moons (1000,2,Noise 10%)	21.10	**19.19**	19.89	24.40	26.30	20.00	19.70
Moons (1000,2,Noise 20%)	23.00	19.79	**19.40**	30.60	30.20	19.50	**19.40**
AVERAGE	18.87	17.68	17.70	20.39	20.74	21.63	**16.92**
UCI							
	KNN	LG	SVM	EE	DGR	KLR	TopoReg
SPECT (267,22)	17.57	17.20	18.68	**16.38**	23.92	18.31	17.54
Congress (435,16)	5.04	4.13	4.59	4.59	4.80	**4.12**	4.58
Molec. (106,57)	24.54	19.10	19.79	17.25	16.32	19.10	**12.62**
Cancer (286,9)	29.36	28.65	28.64	28.68	31.42	29.00	**28.31**
Vertebral (310,6)	15.47	15.46	23.23	17.15	13.56	12.56	**12.24**
Energy (768,8)	0.78	0.65	0.65	0.91	0.78	**0.52**	**0.52**
AVERAGE	15.46	14.20	15.93	14.16	15.13	13.94	**11.80**
Biomedicine							
	KNN	LG	SVM	EE	DGR	KLR	TopoReg
KIRC (243,166)	30.12	28.87	32.56	31.38	35.50	31.38	**26.81**
fMRI (1092,19)	46.70	74.91	74.08	82.51	**31.32**	34.07	33.24

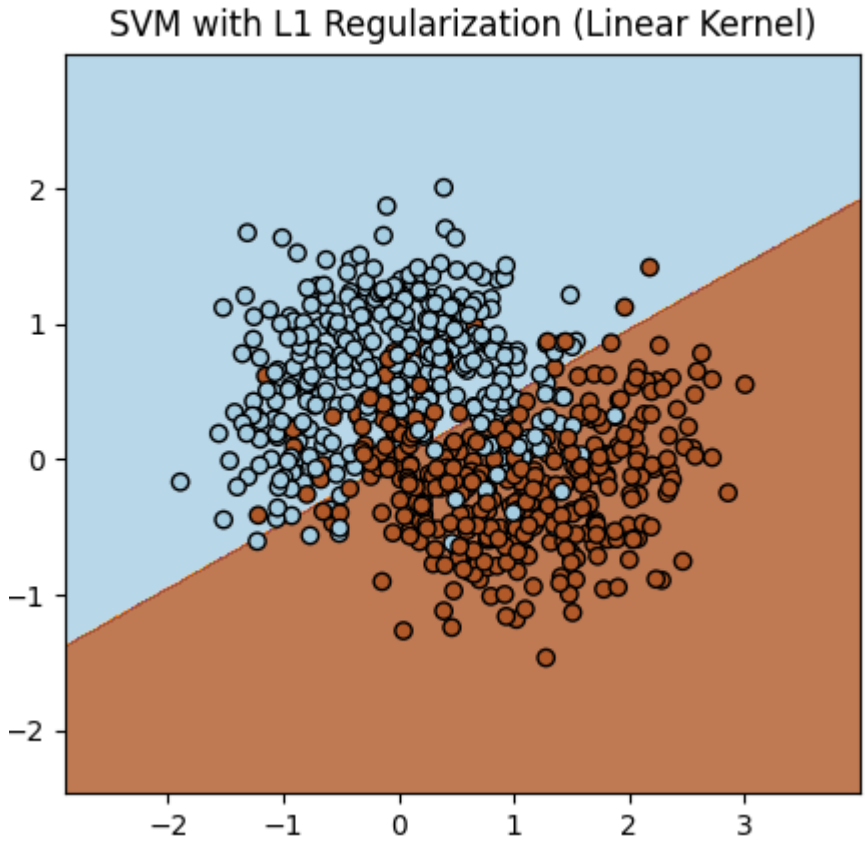
The synthetic data is available in sklearn library and can be easily used for computation and processing.
The UCI datasets can be found in the UCI library online.

OUR RESULTS BASED ON THE ABOVE DATASETS:

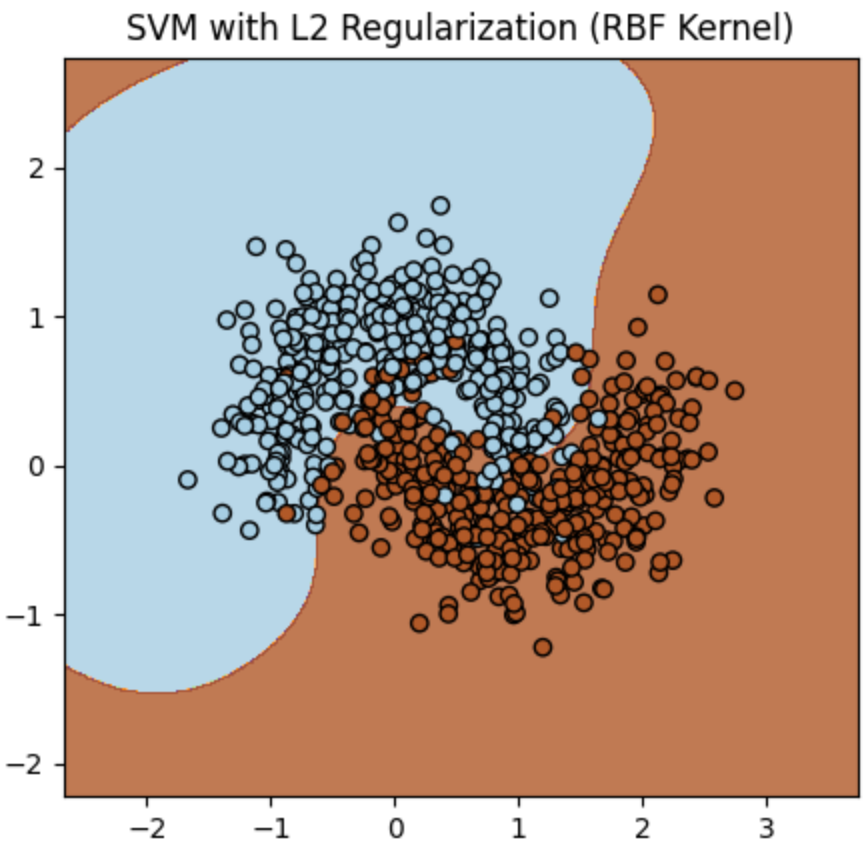
Synthetic Data:

1) **Moons Data**: Moons data contains 1000 points in such a way that the boundary between the classes 0 and 1 is a moon-like crescent structure as evident below. The moons dataset is a synthetic dataset which can be found in the sklearn library.

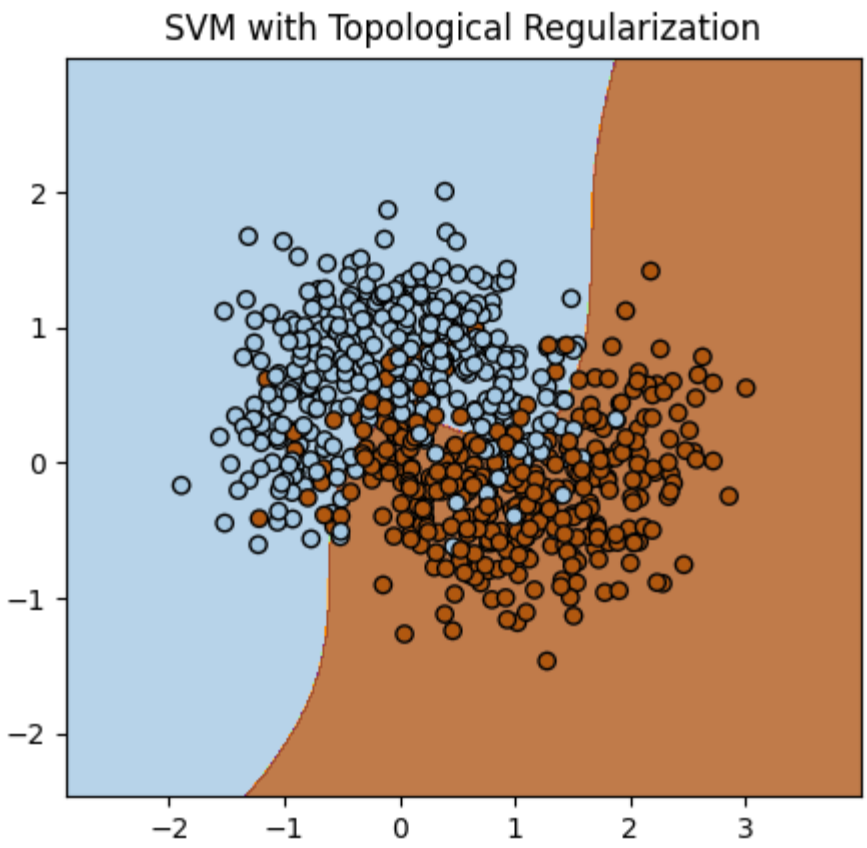
- Moons Data(30% noise)



Training accuracy = 85 %
Testing accuracy = 85.33333333333334 %

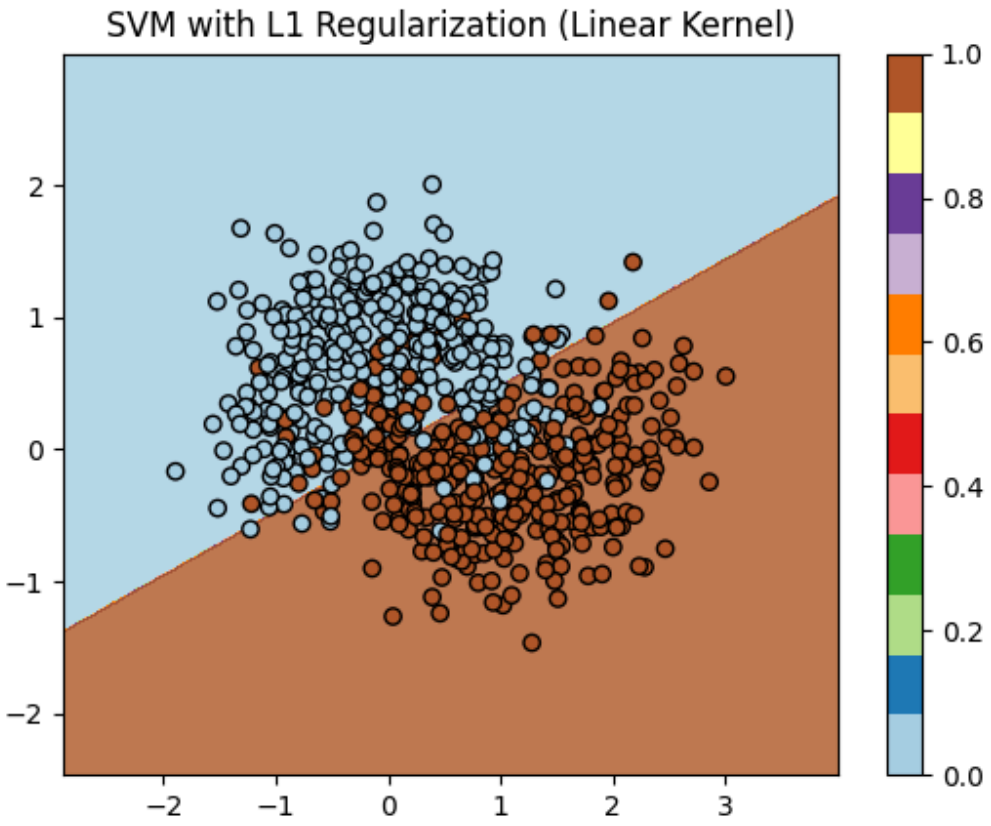


Training accuracy: 0.9142857142857143
Testing accuracy: 0.9033333333333333

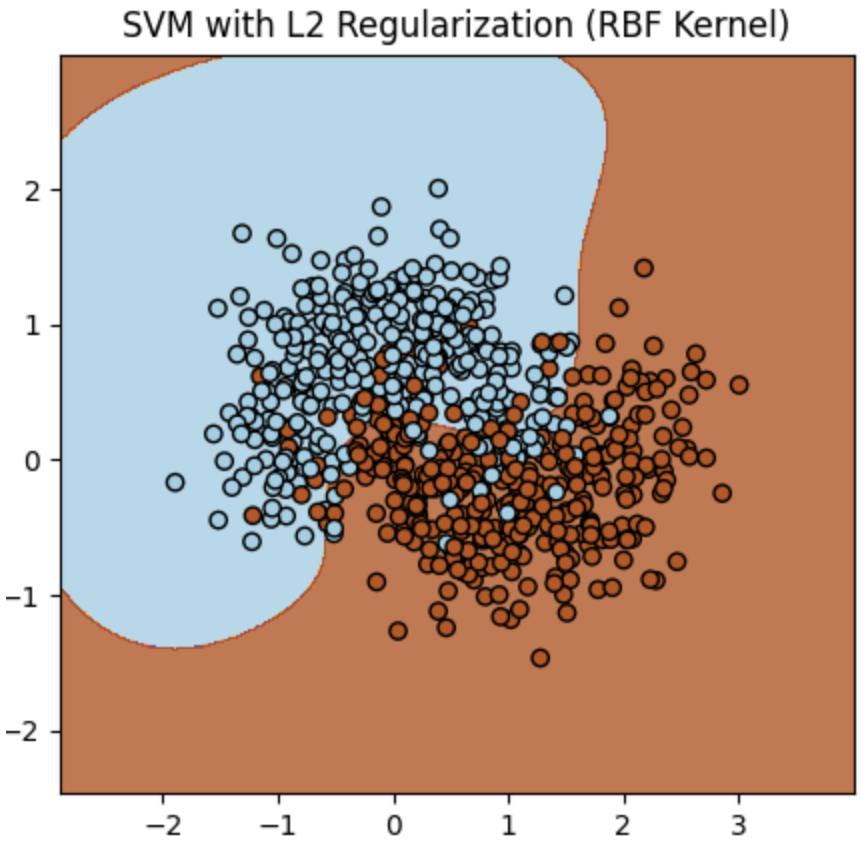


Training accuracy = 86.71 %
Testing accuracy = 86 %

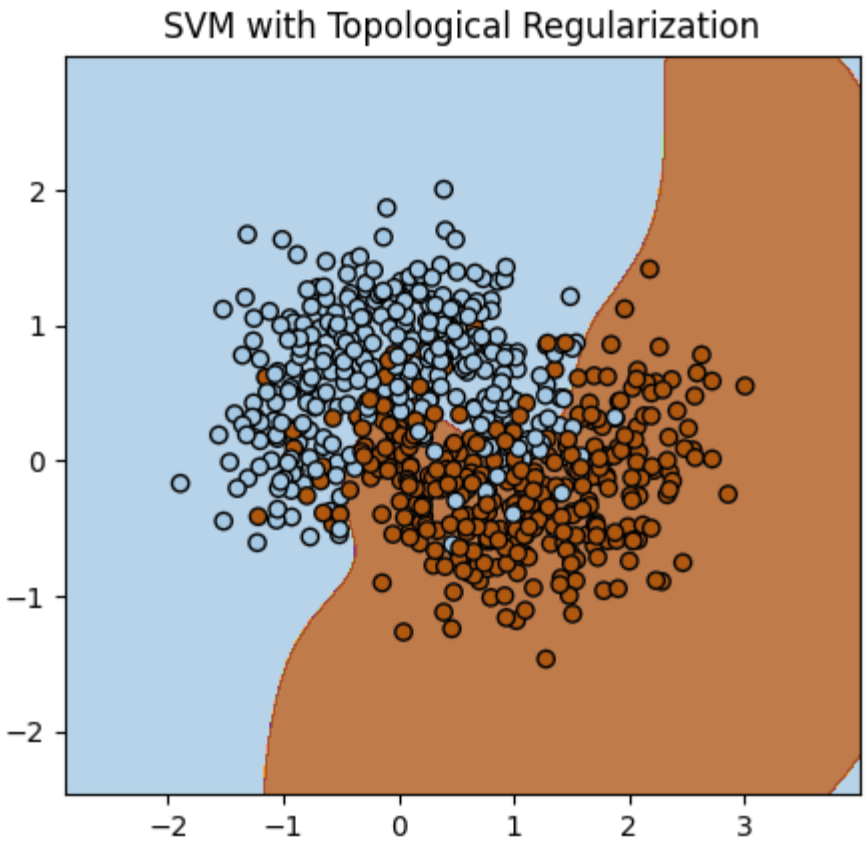
Moons Data (40% noise)



Training accuracy: 83.00 %
Testing accuracy: 83.33333333333334 %

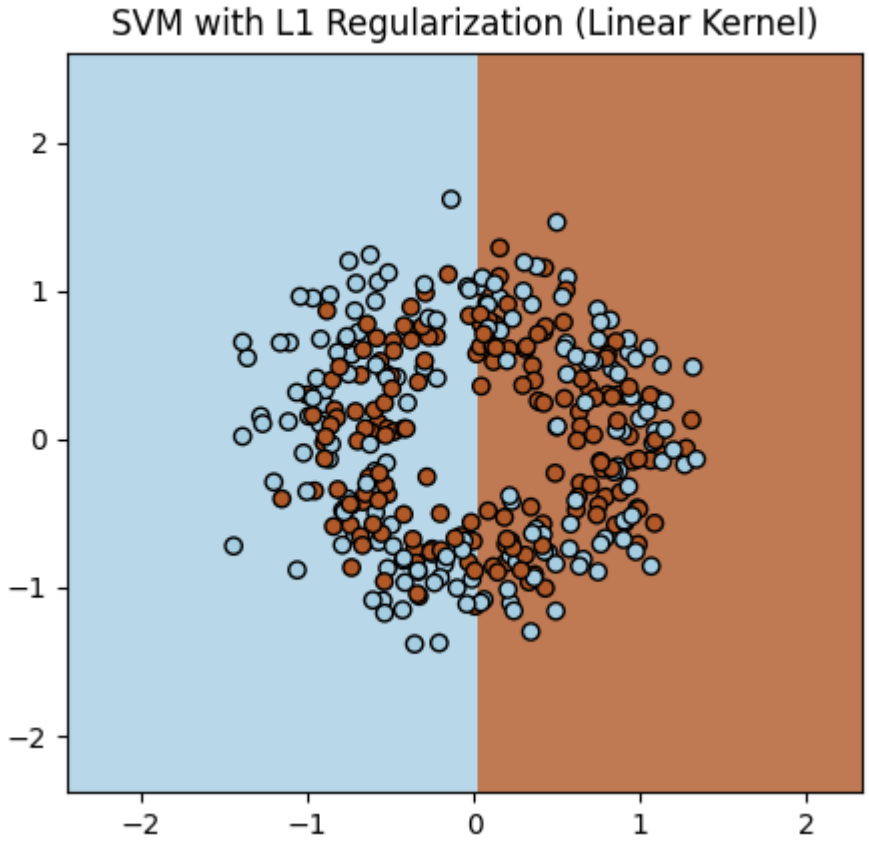


Training accuracy: 86.57142857142858 %
Testing accuracy: 86 %

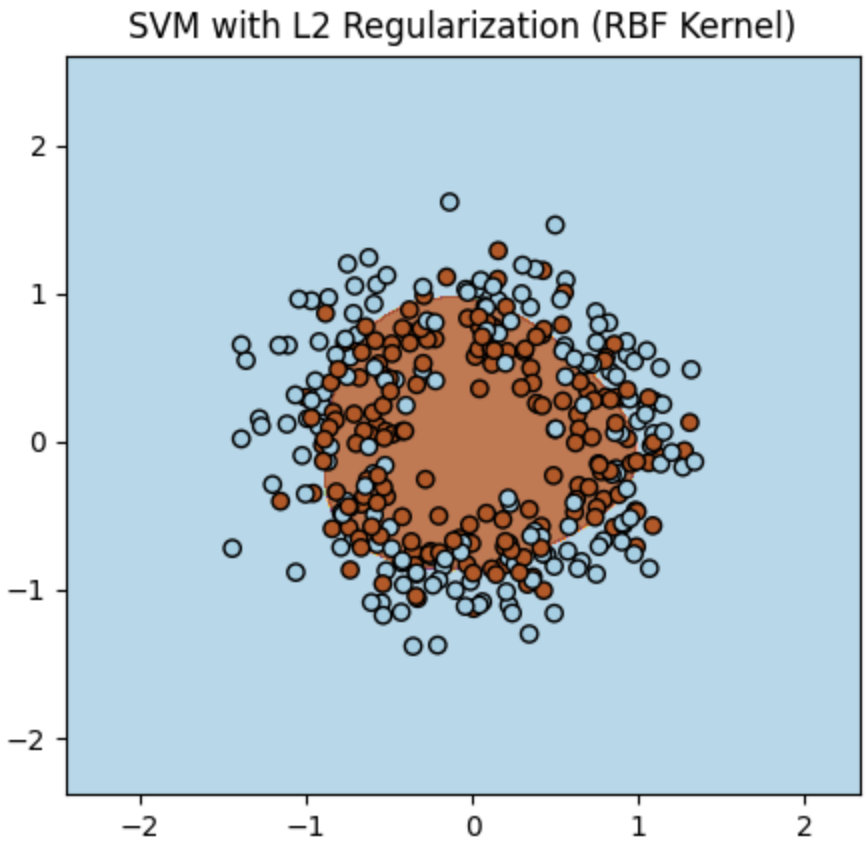


Training accuracy = 87.57 %
Testing accuracy = 85 %

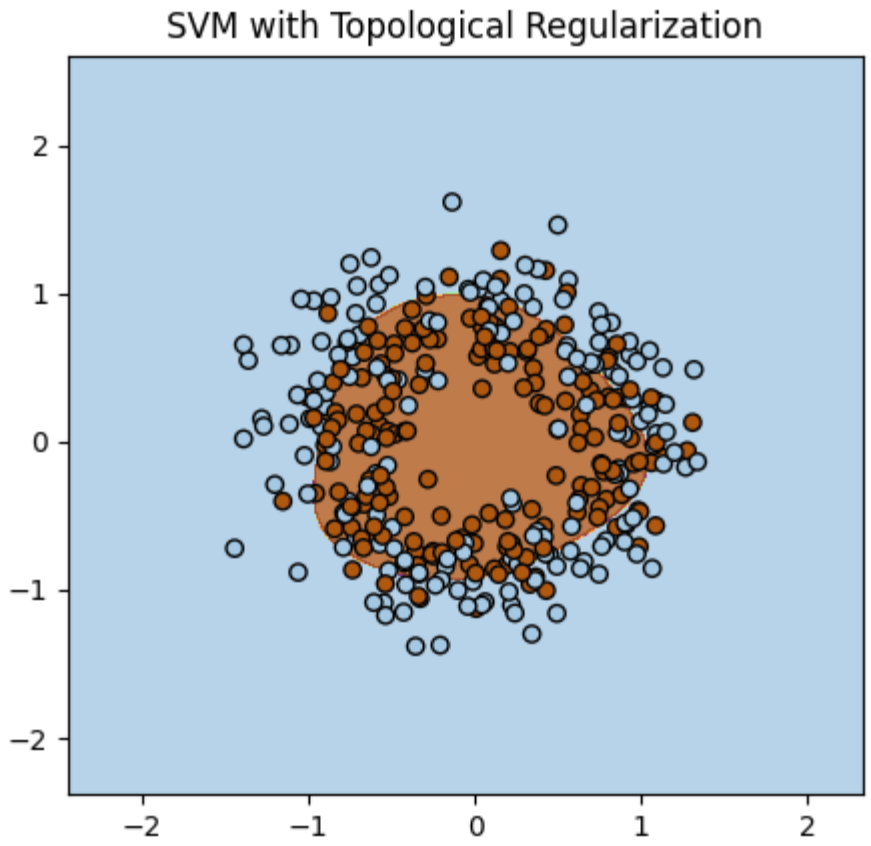
Circles Dataset(20% noise): Circles dataset as shown below consists of 500 data points in such a way that the binary classifier between the classes is a circle based structure.



Training accuracy: 49.714285714285716 %
Testing accuracy: 52.66666666666667 %



Training accuracy: 68.28571428571428 %
Testing accuracy: 66 %



Training accuracy = 70.28 %
Testing accuracy = 66.67 %

Note: For this dataset, the training as well as testing accuracies both are increasing due the introduction of topological penalty because the nature of the data is such that it is prone to overfitting as the number of points are less and also the noise is not very high.

UCI Dataset:

1) **SPECT Dataset:** The dataset describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images. Each of the patients is classified into two categories: normal and abnormal. It contains 267 instances of data having 22 features each.

2) **Congress Dataset:** This dataset includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA. The CQA lists nine different types of votes, based on the type of candidate and opposing candidate.
3) **Breast Cancer Dataset**: This dataset consists of 9 features with 286 instances describing the patient and his/her symptoms, and based on that, they are classifying whether they have cancer or not.

NOTE: In all the above UCI datasets, we have utilised Support Vector Machines(SVM) with rbf kernel, as it was giving high train and test accuracy among all the classifiers.

RESULTS:

SPECT DATASET:

L2 Penalty:
Training Accuracy: 0.8375
Testing Accuracy: 0.727

L1 Penalty:
Training Accuracy: 0.85
Testing Accuracy: 0.770

With Topological Penalty:
Training accuracy: 0.8127
Testing accuracy : 0.7754

Congress DATASET:

Note: Congress dataset doesn't perform well with SVM classifier, instead it performs better with logistic regression with multi-label classification compression.
Now, L1 and L2 regularisation doesn't make sense in case of logistic regression, since there is no overfitting case. Thus, the below outputs are of SVM which are suboptimal, but does stand with our logic and use of topological penalty.

L2 Penalty:
Training Error Rate: 0.02011
Testing Error Rate: 0.0804

L1 Penalty:
Training Error Rate: 0.01436
Testing Error Rate: 0.06896

With Topological Penalty:
Training Error Rate: 0.00862
Testing Error Rate: 0.06896

Breast Cancer DATASET:

L2 Penalty:
Train Accuracy: 0.754
Test Accuracy: 0.689
Mean Error Rate: 0.3103

L1 Penalty:
Train Accuracy: 0.5701
Test Accuracy: 0.5689
Mean Error Rate: 0.4310

With Topological Penalty:
Train Accuracy: 0.7894
Test Accuracy: 0.6379
Mean Error Rate: 0.3620

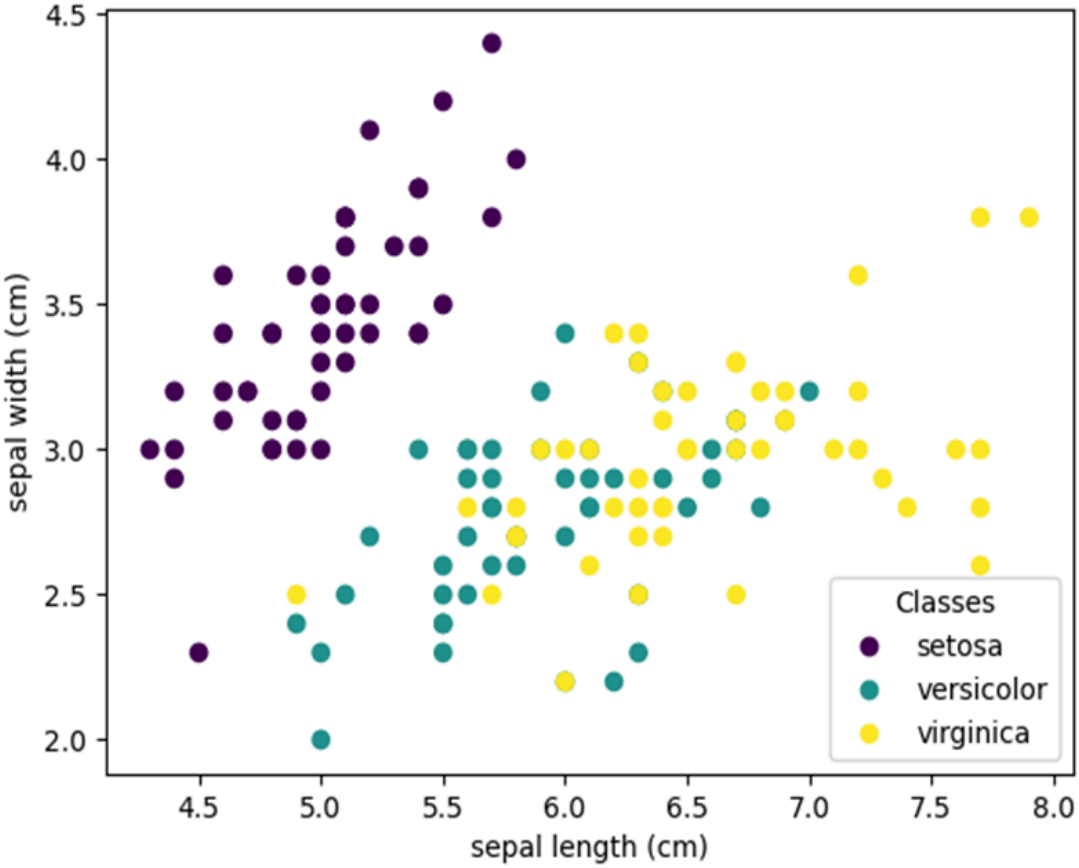
Note: Breast Cancer dataset performs better with L2 regularisation than topological penalty. This might be due to the dataset properties which makes L2 regularisers more suitable.

Since, these datasets had multiple columns, it was difficult to visualize and gain inferences from data plots.

Multi-label classification:

The paper suggested trying our model and regulariser for multi-label classification as part of future work. Thus, we have implemented two datasets for multi-label classification, as part of extending the scope of the paper. These datasets can be found in the UCI library.

1. **Iris Dataset**:The Iris dataset is a well-known dataset in the field of machine learning and statistics. It comprises 150 samples, each describing an iris flower from one of three species: setosa, versicolor, and virginica. Each sample includes four features: sepal length, sepal width, petal length, and petal width. These features, measured in centimeters, are used to classify the species of the iris flower. With its 4-dimensional feature space and three distinct classes, the Iris dataset serves as a classic example for various multi-label classification algorithms.



Results(Using SVM(rbf) kernel):

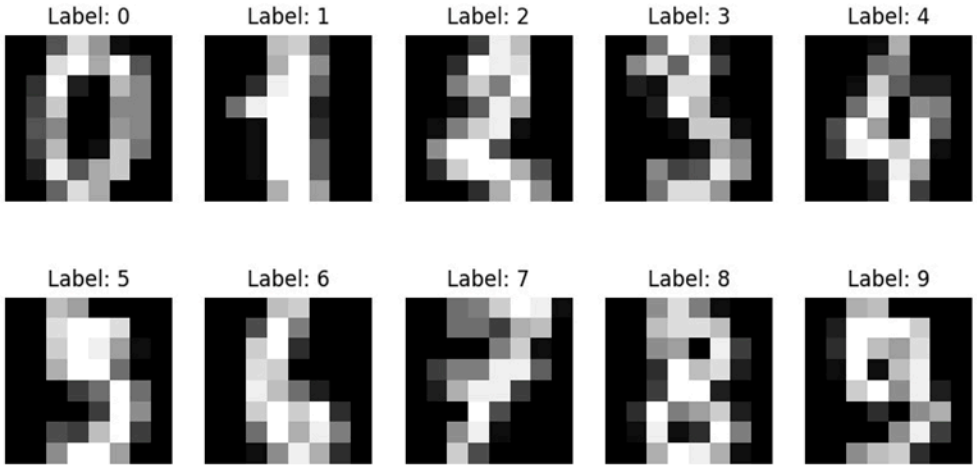
Training accuracy without topological penalty: 0.97037

Testing accuracy without topological penalty: 0.93333

Training accuracy with topological penalty: 0.96296

Testing accuracy with topological penalty: 0.95433

2. **Load Digits Dataset:** The Load Digits dataset is widely utilized in the fields of image recognition and machine learning. It contains 1,797 samples of handwritten digits, each represented as an 8x8 pixel image. Each image is flattened into a 64-dimensional feature vector, capturing the intensity of each pixel. The dataset includes digits from 0 to 9, resulting in ten distinct classes. This dataset is often employed to train and evaluate classification algorithms, providing a robust test bed for developing models that can accurately recognize and distinguish between different handwritten digits.



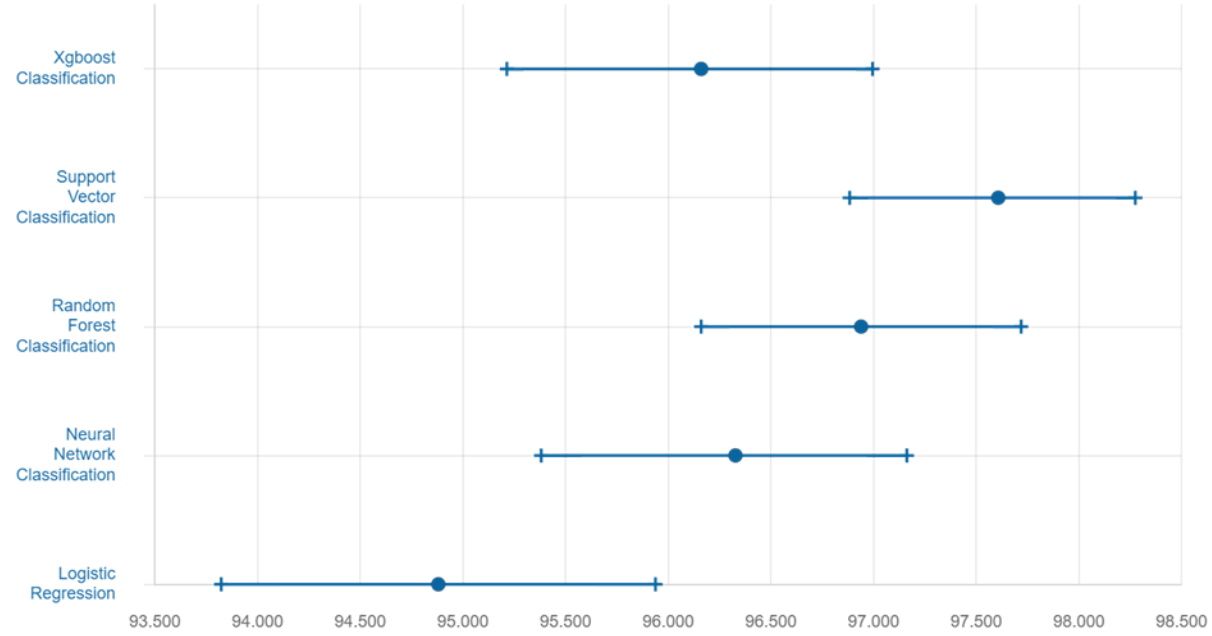
Results:

Training accuracy without topological penalty: 0.99363

Testing accuracy without topological penalty: 0.97407

Training accuracy with topological penalty: 0.9833

Testing accuracy with topological penalty: 0.98148

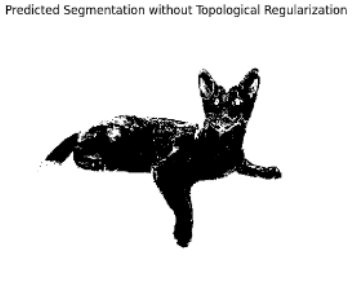


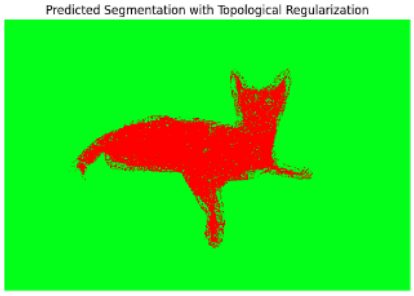
The above image shows the performance of various classifiers for the load digits dataset.

IMAGE SEGMENTATION RESULTS

Dataset Used: Oxford Pets Dataset
Model chosen for regularization: Random Forest Classifier

Accuracy without topological penalty = 0.95
Accuracy with topological penalty = 0.96





Accuracy without topological penalty = 0.96
Accuracy with topological penalty = 0.97

Note: The aim of the algorithm was to prevent overfitting of the model by removing less robust small and noisy connected components and the same thing is visible in the image segmentation results also where small noisy white dots (less robust components) have been removed and accuracy has increased.

CONCLUSION:

Topological regularizers are a great way of increasing test accuracies on data, and thus, it can be suggested as a new way to regularize the models like Lasso and Ridge regularisers. We can also navigate the concepts of unsupervised classification using the concept of topological gradient, as utilized above in our method.

The above method can also find it’s use in the field of image segmentation by utilizing persistence pairings for implementing persistence diagram, which is our follow-up idea.