
SEIS763 Machine Learning - Write a program to perform SVM using RBF kernel

Table of Contents

Clear history, Load dataset	1
Define Target	1
Define XY Matrix	1
Build SVM RBF Kernel Model	1
Predict for classes 0 and 1	2
Q&A	2

Assignment 6

Garth Mortensen, mort0052@stthomas.edu

Clear history, Load dataset

```
% clear all
% clc
% close all
CellDNA = csvread('C:\tmp\CellDNA.csv');
```

Define Target

```
Y = CellDNA(:,14:14);
%Convert this dependent variable to binary values.
%Y = 0 means bacterium is not interesting to study
Y(Y ~= 0) = 1;
```

Define XY Matrix

```
XAll = CellDNA(:,1:13);
% Standardize numerics
XAll = zscore(XAll);
XAllY = [XAll Y];

% [train_XAllY, test_XAllY] = dividerand(XAllY.', 0.7, 0.3);
% XAllY = XAllY;
% How the heck do i train test split?
```

Build SVM RBF Kernel Model

produce a model Model for Y = 0

```
YZero = -1*(Y - 1);
classificationSVMZero = fitcsvm(XAllY(:, 1:13),
    YZero, 'KernelFunction', 'RBF');

classificationSVM = fitcsvm(XAllY(:, 1:13), XAllY(:,
    14:14), 'KernelFunction', 'RBF');
```

Predict for classes 0 and 1

```
[class0 score0] = predict(classificationSVMZero, XAllY(:, 1:13));
[class1 score1] = predict(classificationSVM, XAllY(:, 1:13));
```

Q&A

1. What is the accuracy, Precision, and Recall for each class prediction?

- Class 0

```
C0 = confusionmat(YZero, classificationSVMZero.Y)
TP0 = C0(1,1);
TN0 = C0(2,2);
FP0 = C0(1,2);
FN0 = C0(2,1);
precision0 = TP0/(TP0 + FP0);
recall0 = TP0/(TP0 + FN0);
fscore0 = 2*((precision0 * recall0)/(precision0 + recall0));
```

C0 =

200	0
0	1017

- Class 1

```
C1 = confusionmat(Y, classificationSVM.Y)
TP1 = C1(1,1);
TN1 = C1(2,2);
FP1 = C1(1,2);
FN1 = C1(2,1);
precision1 = TP1/(TP1 + FP1);
recall1 = TP1/(TP1 + FN1);
fscore1 = 2*((precision1 * recall1)/(precision1 + recall1));
```

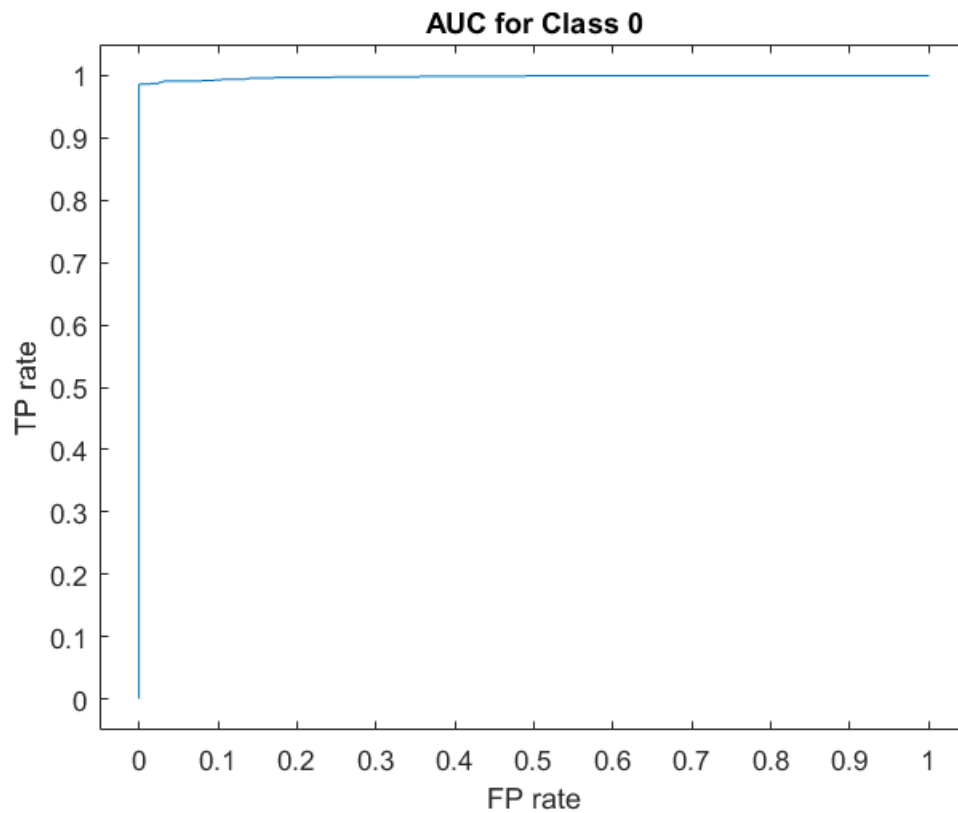
C1 =

1017	0
0	200

2. Create an ROC curve plot for each class prediction.

- Class 0

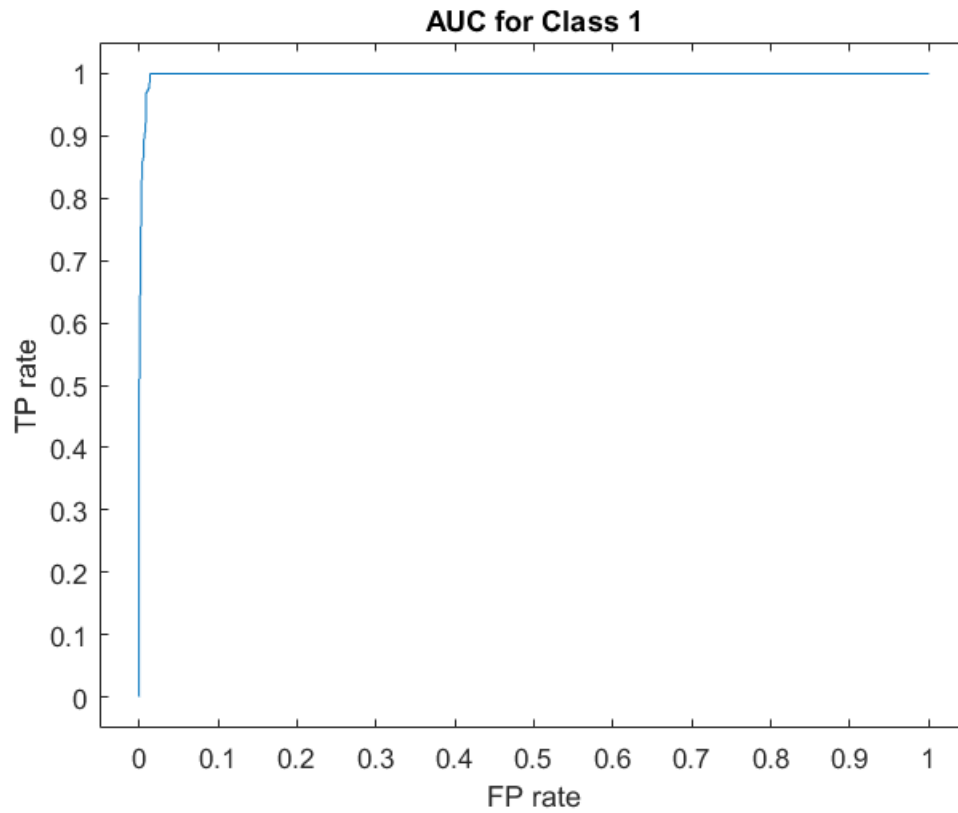
```
[xpos, ypos, T, AUC0] = perfcurve(Y, score1(:, 1), 0); %2 columns in score
figure, plot(xpos, ypos)
xlim([-0.05 1.05]), ylim([-0.05 1.05])
xlabel('FP rate'), ylabel('TP rate')
title('AUC for Class 0')
```



- Class 1

```
[xpos, ypos, T, AUC1] = perfcurve(Y, score1(:, 2), 1); %2 columns in score
figure, plot(xpos, ypos)
xlim([-0.05 1.05]), ylim([-0.05 1.05])
xlabel('FP rate'), ylabel('TP rate')
title('AUC for Class 1')
```

%Future - do a train-test split



Published with MATLAB® R2016a