

Contents

- This scripts trains and test the SVM classifier on a sample data set
- 1 - Load training data
- 2 - Extract 2D training features (standard deviation and convolution ratio)
- 3 - Train the SVM classifier using the extracted features
- 4 - Plot trained classifier
- 5 - Load testing data
- 6 - Extract 2D testing features (standard deviation and convolution ratio)
- 7 - Use the trained SVM classifier to make label predictions on testing data
- 8 - Plot the SVM classificaiton performance on the testing data

This scripts trains and test the SVM classifier on a sample data set

1 - Load training data

```
datapath = '/Users/akv020/Projects/ML_classification_of_dust/GitHub/Train_and_test_data/';
X_train = csvread([datapath,'Train_data.csv']);
Y_train = csvread([datapath,'Train_labels.csv']);
```

2 - Extract 2D training features (standard deviation and convolution ratio)

```
features_train = extract_SVM_features(X_train);
```

3 - Train the SVM classifier using the extracted features

```
labels_train = (2*Y_train)-1;
clSVM = fitcsvm(features_train,labels_train','KernelFunction','polynomial','PolynomialOrder',2);
```

4 - Plot trained classifier

```
FIG = figure('units','centimeters','position',[1,1,25.0,24.0]);
sx = 0.08;
sy = 0.085;
fz = 18;
mz = 5;
lw = 3;
d = 0.01;
x1 = 0;
x2 = 0.5;
y1 = 0;
y2 = 4;

% Make grid of data points
[x1Grid,x2Grid] = meshgrid((0:d:0.5),(0:d:4));
xGrid = [x1Grid(:),x2Grid(:)];

%Predict scores over the grid
[label,scores,costs] = predict(clSVM,xGrid);

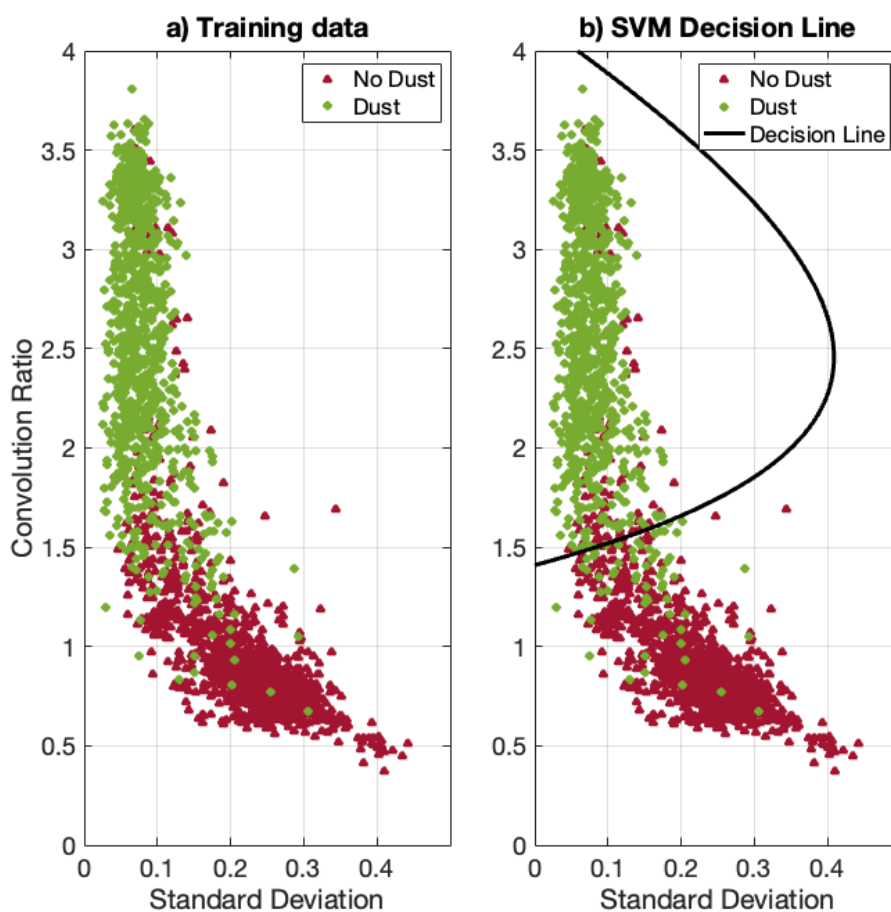
% Plot the data
subplot_tight(1,2,1,[sx,sy])
h = gscatter(features_train(:,1),features_train(:,2),labels_train,'rg','^o',mz,'filled');
xlabel('Standard Deviation')
ylabel('Convolution Ratio')
legend({'No Dust','Dust'},'Location','northeast');
grid on
xlim([x1 x2])
ylim([y1 y2])
set(gca,'fontsize',fz)
title('a) Training data')
h(1).Color = [0.6350 0.0780 0.1840];
h(2).Color = [0.4660 0.6740 0.1880];
```

```

h(1).MarkerFaceColor = [0.6350 0.0780 0.1840];
h(2).MarkerFaceColor = [0.4660 0.6740 0.1880];

% Plot the data with the SVM decision boundary
subplot_tight(1,2,2,[sx,sy])
h = gscatter(features_train(:,1),features_train(:,2),labels_train,'rg','^o',mz,'filled');
hold on
xlim([x1 x2])
ylim([y1 y2])
contour(x1Grid,x2Grid,reshape(scores(:,2),size(x1Grid)),[0 0],'k','LineWidth',lw);
xlabel('Standard Deviation')
h(1).Color = [0.6350 0.0780 0.1840];
h(2).Color = [0.4660 0.6740 0.1880];
h(1).MarkerFaceColor = [0.6350 0.0780 0.1840];
h(2).MarkerFaceColor = [0.4660 0.6740 0.1880];
legend({'No Dust','Dust','Decision Line'},'Location','northeast');
grid on
set(gca,'fontsize',fz)
title('b) SVM Decision Line')

```



5 - Load testing data

```

X_test = csvread([datapath,'Test_data.csv']);
Y_test = csvread([datapath,'Test_labels.csv']);

```

6 - Extract 2D testing features (standard deviation and convolution ratio)

```

features_test = extract_SVM_features(X_test);

```

7 - Use the trained SVM classifier to make label predictions on testing data

```

labels_test = (2*Y_test)-1;
[SVM_labels] = predict(c1SVM,[features_test(:,1),features_test(:,2)]);

```

8 - Plot the SVM classification performance on the testing data

```

FIG = figure('units','centimeters','position',[1,1,36.0,24.0]);
sx = 0.08;
sy = 0.06;
fz = 18;
mkz = 50;
lw = 3;
d = 0.01;

% Plot the testing data
subplot_tight(1,3,1,[sx,sy])
scatter(features_test(:,1),features_test(:,2),mkz,'k','square','filled');
xlabel('Standard Deviation')
ylabel('Convolution Ratio')
legend({'Test Data'},'Location','northeast');
grid on
xlim([x1 x2])
ylim([y1 y2])
set(gca,'fontsize',fz)
title('a) Testing Data')
xticks([0 0.1 0.2 0.3 0.4])

% Plot the testing data and the SVM label predictions
subplot_tight(1,3,2,[sx,sy])
h = gscatter(features_test(:,1),features_test(:,2),SVM_labels,'rg','^o',mz,'filled');
hold on
contour(x1Grid,x2Grid,reshape(scores(:,2),size(x1Grid)),[0 0],'k','LineWidth',lw);
xlabel('Standard Deviation')
legend({'No Dust','Dust','Decision Line'},'Location','northeast');
grid on
xlim([x1 x2])
ylim([y1 y2])
set(gca,'fontsize',fz)
title('b) SVM Classification')
h(1).Color = [0.6350 0.0780 0.1840];
h(2).Color = [0.4660 0.6740 0.1880];
h(1).MarkerFaceColor = [0.6350 0.0780 0.1840];
h(2).MarkerFaceColor = [0.4660 0.6740 0.1880];
xticks([0 0.1 0.2 0.3 0.4])

cp = classperf(1-labels_test,1-SVM_labels);
accuracy = round(cp.CorrectRate,2);

% Plot the testing data and the "true" labels
subplot_tight(1,3,3,[sx,sy])
h = gscatter(features_test(:,1),features_test(:,2),labels_test,'rg','^o',mz,'filled');
hold on
contour(x1Grid,x2Grid,reshape(scores(:,2),size(x1Grid)),[0 0],'k','LineWidth',lw);
xlabel('Standard Deviation')
legend({'No Dust','Dust','Decision Line'},'Location','northeast');
grid on
xlim([x1 x2])
ylim([y1 y2])
set(gca,'fontsize',fz)
title(['c) Manual Labels, Accuracy: ', num2str(accuracy)])
h(1).Color = [0.6350 0.0780 0.1840];
h(2).Color = [0.4660 0.6740 0.1880];
h(1).MarkerFaceColor = [0.6350 0.0780 0.1840];
h(2).MarkerFaceColor = [0.4660 0.6740 0.1880];
xticks([0 0.1 0.2 0.3 0.4])

```

