**CSCE 623 Spring 2020 - Machine Learning. In Class Work, Day 16**

From Chapter 9: Support Vector Machines

For the purposes of this exercise, you may assume all dataframes have 2 real number features (‘X1’,’X2’) and a real number output label (‘Y’). The data will have *n* observations. In this exercise, you will write code explore SVMs performance on various datasets with different kernels and margin penalties. Some starter code is provided on blackboard

0. **(Provided by Instructor)** Import a bunch-of-python-goodness:

import numpy as np

import matplotlib.pyplot as plt

from sklearn import svm

from sklearn import datasets

1. **(Provided by Instructor)** Using sklearn.datasets (<http://scikit-learn.org/stable/modules/classes.html#module-sklearn.datasets>), make a dataset using the make\_blob method call For example:

sampleSeed = 1

dsetSize = 200

#generate blobs which are not quite linearly separable

X,y = datasets.make\_blobs(dsetSize,2,centers = [[-1.2,-1.2],[1.2,1.2]],

random\_state = sampleSeed)

2. **(Provided by Instructor)** Fit a support vector classification model using SVC and a ‘linear’ kernel:

modelSVCl = svm.SVC(kernel = 'linear', C=1)

modelSVCl.fit(X,y)

3. **(Provided by Instructor)** A simple decision region visualizer is provided. For example, this one plots the datapoints, the decision boundary, and the prediction colors on each side of the boundary. :

def drawDecisionBoundary(model,X,title):

x\_min, x\_max = X[:, 0].min() - 1, X[:, 0].max() + 1

y\_min, y\_max = X[:, 1].min() - 1, X[:, 1].max() + 1

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, 0.1),

np.arange(y\_min, y\_max, 0.1))

#fig = plt.subplots(1, 1, sharex='col', sharey='row', figsize=(10, 8))

plt.figure(figsize=(10, 8))

Z = model.predict(np.c\_[xx.ravel(), yy.ravel()])

Z = Z.reshape(xx.shape)

plt.contourf(xx,yy, Z, alpha = 0.4, cmap="binary")

plt.scatter(X[:, 0], X[:, 1], c=-y, cmap="jet", alpha=0.6)

plt.title(title)

4. **STUDENT CODE MODIFICATION**: Add capability to your decision region visualizer to mark the support vectors on the plot from your model (using model.support\_ )

5. **(Provided by Instructor)** Use the decision boundary function from 4 to plot out your decision boundary of your model when fit to the data. Explore how altering the dataset and altering C can change the resulting support vectors.

6. **STUDENT EXPLORATION**: Fit several different support vector classification models, including SVC **polynomial** and **rbf** kernels to the data. Examine how the boundaries change. Try different hyperparameter settings

7. **STUDENT CODE MODIFICATION**: Explore datasets by selecting the global variable datasettype = BLOBS (or CONCENTRIC) and perhaps repeat the model fitting with different SVC kernels and hyperparameter values.