

ECE469 Homework 3

Chase A. Lotito

December 4, 2024

Question 1

Train SVM classifiers using a Gaussian Kernel based on MATLAB.

Solution.

```
1 %{\n2 =====\n3 Homework 3 Question 1 Code\n4 Name      : Chase Lotito\n5 University : Southern Illinois University\n6 Course    : ECE469\n7 =====\n8 Description:\n9 Training a SVM classifiers using a\n10 Gaussian Kernel based on MATLAB\n11 =====\n12 %}\n13\n14 % (A) GENERATE 200 POINTS IN UNIT DISK\n15 rng(1);\n16 r = sqrt(rand(200,1));\n17 theta = 2*pi*rand(200,1);\n18 data1 = [r.*cos(theta), r.*sin(theta)];\n19\n20 % (B) GENERATE 200 POINTS IN ANNULUS (washer)\n21 r2 = sqrt(3*rand(200,1)+1);\n22 theta2 = 2*pi*rand(200,1);\n23 data2 = [r2.*cos(theta2), r2.*sin(theta2)];\n24\n25 % (C) PLOT data1, data2, AND CIRCLES OF RADIUS 1 AND 2\n26 figure;\n27 plot(data1(:,1), data1(:,2), 'r.', 'MarkerSize', 15)\n28 hold on\n29 plot(data2(:,1), data2(:,2), 'b.', 'MarkerSize', 15)\n30 ezpolar(@(x)1);\n31 ezpolar(@(x)2);\n32 axis equal\n33 hold off\n34\n35 % (D) COMBINE data1 AND data2 INTO ONE MATRIX TO GENERATE\n36 %     A VECTOR FOR CLASSIFICATIONS\n37 data3 = [data1; data2];
```

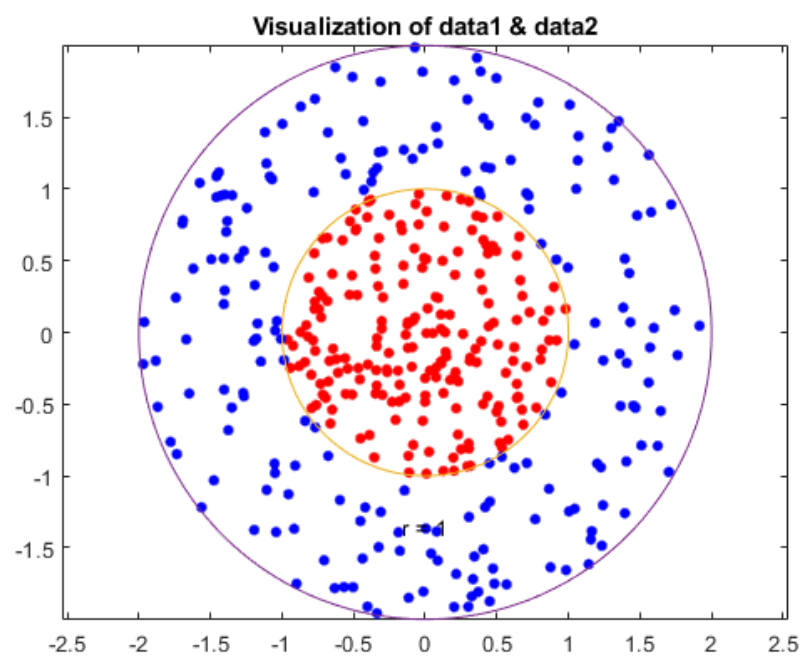
```

38 theclass = ones(400,1); % 400x1 vector full of 1s
39 theclass(1:200) = -1; % set first 200 1s to -1s
40
41 % (E) TRAIN A SVM CLASSIFIER WITH "KernelFunction" SET TO
42 % "rbf" AND BoxConstraint SET TO Inf. PLOT THE DECISION
43 % BOUNDARY AND FLAG THE SUPPORT VECTORS
44
45 % train svm
46 cl = fitcsvm(data3, theclass, 'KernelFunction', 'rbf', 'BoxConstraint', 1,
47 'ClassNames', [-1,1]);
48
49 % predict scores over grid
50 d = 0.02;
51 [x1Grid, x2Grid] = meshgrid(min(data3(:,1)):d:max(data3(:,1)),min(data3
52 (:,2)):d:max(data3(:,2)));
53 xGrid = [x1Grid(:), x2Grid(:)];
54 [~, scores] = predict(cl, xGrid);
55
56 % plot data and decision boundary
57 figure;
58 h(1:2) = gscatter(data3(:,1), data3(:,2), theclass, 'rb', '.');
59 hold on
60 ezpolar(@(x)1);
61 h(3) = plot(data3(cl.IsSupportVector,1), data3(cl.IsSupportVector,2), 'ko',
62 );
63 contour(x1Grid, x2Grid, reshape(scores(:,2), size(x1Grid)), [0 0], 'k');
64 legend(h, {'-1', '+1', 'Support Vectors'});
65 axis equal
66 hold off

```

Code 1: q1.m

Code 1 utilizes a user-defined function to implement the Gaussian Kernel which can be referenced in Code 6.



Question 2

Solution.

```
1 %{\n2 =====\n3 Homework 3 Question 2 Code\n4 Name      : Chase Lotito\n5 University : Southern Illinois University\n6 Course    : ECE469\n7 =====\n8 Description:\n9 Training a SVM classifiers using a custom\n10 Kernel (Sigmoid kernel) based on MATLAB\n11 =====\n12 %}\n13\n14 % (A) GENERATE RANDOM SET OF POINTS IN UNIT CIRCLE.\n15 %     Q1 AND Q3 POINTS POSITIVE CLASS.\n16 %     Q2 AND Q4 POINTS NEGATIVE CLASS.\n17\n18 rng(1);\n19 n = 100;\n20\n21 r1 = sqrt(rand(2*n,1));\n22 t1 = [pi/2*rand(n,1) ; (pi/2*rand(n,1)+pi)];\n23 X1 = [r1.*cos(t1) r1.*sin(t1)];\n24\n25 r2 = sqrt(rand(2*n,1));\n26 t2 = [pi/2*rand(n,1) + pi/2 ; (pi/2*rand(n,1) - pi/2)];\n27 X2 = [r2.*cos(t2) r2.*sin(t2)];\n28\n29 X = [X1 ; X2];\n30 Y = ones(4*n,1);\n31 Y(2*n + 1:end) = -1;\n32\n33 % (B) PLOT DATAPOINTS\n34 figure;\n35 gscatter(X(:,1),X(:,2),Y);\n36 title('Scatter Diagram of Simulated Data');\n37\n38 % (C) USING THE SIGMOID KERNEL. WRITE A TRANSFORMATION\n39 %     FUNCTION WHICH GENERATES THE GRAM MATRIX GIVEN\n40 %     TWO MATRICIES AND THE SIGMOID KERNEL\n41\n42 % DEFINED IN SEPARATE FUNCTION FILES\n43\n44 % (D) TRAIN SVM CLASSIFIER USING SIGMOID KERNEL\n45 Mdl1 = fitcsvm(X, Y, 'KernelFunction', 'mysigmoid', 'Standardize',true);\n46\n47 % (E) PLOT DATA. IDENTIFY SUPPORT VECTORS AND BOUNDARY\n48 % predict scores over grid\n49 d = 0.02;\n50 [x1Grid, x2Grid] = meshgrid(min(X(:,1)):d:max(X(:,1)),min(X(:,2)):d:max(X
```

```

        (:,2)));
51 xGrid = [x1Grid(:), x2Grid(:)];
52 [~, scores] = predict(Mdl1, xGrid);
53
54 % plot data and decision boundary
55 figure;
56 h(1:2) = gscatter(X(:,1), X(:,2),Y);
57 hold on
58 ezpolar(@(x)1);
59 h(3) = plot(X(Mdl1.IsSupportVector,1), X(Mdl1.IsSupportVector,2), 'ko', '
        MarkerSize',10);
60 contour(x1Grid, x2Grid, reshape(scores(:,2), size(x1Grid)), [0 0], 'k');
61 title('Scatter Diagram with the Decision Boundary');
62 legend({'-1', '+1', 'Support Vectors'}, 'Location', 'Best');
63 hold off
64
65 % (F) DETERMINE OUT-OF-SAMPLE MISCLASSIFICATION RATE
66 %     USING 10-FOLD CROSS VALIDATION
67 CVMdl = crossval(Mdl1);
68 misclass1 = kfoldLoss(CVMdl);
69 misclass1
70
71 % (G) WRITE NEW SIGMOID KERNEL W/ gamma=0.5 AND RETRAIN
72
73 % (D) TRAIN SVM CLASSIFIER USING SIGMOID KERNEL
74 Mdl2 = fitcsvm(X, Y, 'KernelFunction', 'mysigmoid_2', 'Standardize',true);
75
76 % (E) PLOT DATA. IDENTIFY SUPPORT VECTORS AND BOUNDARY
77 % predict scores over grid
78 d = 0.02;
79 [x1Grid, x2Grid] = meshgrid(min(X(:,1)):d:max(X(:,1)),min(X(:,2)):d:max(X
        (:,2)));
80 xGrid = [x1Grid(:), x2Grid(:)];
81 [~, scores] = predict(Mdl2, xGrid);
82
83 % plot data and decision boundary
84 figure;
85 h(1:2) = gscatter(X(:,1), X(:,2),Y);
86 hold on
87 ezpolar(@(x)1);
88 h(3) = plot(X(Mdl2.IsSupportVector,1), X(Mdl2.IsSupportVector,2), 'ko', '
        MarkerSize',10);
89 contour(x1Grid, x2Grid, reshape(scores(:,2), size(x1Grid)), [0 0], 'k');
90 title('Scatter Diagram (Model 2) with the Decision Boundary');
91 legend({'-1', '+1', 'Support Vectors'}, 'Location', 'Best');
92
93 CVMdl2 = crossval(Mdl2);
94 misclass2 = kfoldLoss(CVMdl2);
95 misclass2

```

Code 2: q2.m

Question 3

Solution.

```
1 %{\n2 =====\n3 Homework 3 Question 3 Code\n4 Name      : Chase Lotito\n5 University : Southern Illinois University\n6 Course    : ECE469\n7 =====\n8 Description:\n9 Design of a support vector machine for an\n10 object recognition system\n11 =====\n12 %}\n13\n14 % Load in dataset from ../svm/dataset2.mat\n15 % Dataset is a struct of X (100x4)-matrix\n16 % and Y (100x1)-vector\n17 dataset2 = load('C:\\Users\\cloti\\OneDrive\\Desktop\\SCHOOL\\SIUC\\f24\\ece469\\hw\n      \\hw3\\svm\\svm\\dataset2.mat');\n18\n19 % Extract input features X and output target y\n20 X = dataset2.X;\n21 y = dataset2.Y; % y = {1, 0}\n22\n23\n24 % visualize dataset via heatmap\n25 figure;\n26 correlationMatrix = corr(X);\n27 heatmap(correlationMatrix, 'Colormap', jet, 'ColorbarVisible', 'on');\n28 title('Correlation Matrix of Features');\n29\n30 % Train SVM Classifier\n31 % ((1) Linear Kernel)\n32 mdl1 = fitcsvm(X, y, 'KernelFunction', 'linear');\n33 % ((2) Polynomial Kernel)\n34 mdl2 = fitcsvm(X, y, 'KernelFunction', 'polynomial', 'PolynomialOrder', 2);\n35 % ((3) Guassian Kernel)\n36 mdl3 = fitcsvm(X, y, 'KernelFunction', 'mysigmoid_2', 'Standardize', true);\n37\n38 % Cross-validate model with 4-fold cross-validation\n39 cvMdl1 = crossval(mdl1, 'KFold', 4); % Linear 4-fold cross-val\n40 misclas1 = kfoldLoss(cvMdl1); % calc losses\n41 cvMdl2 = crossval(mdl2, 'KFold', 4); % Poly 4-fold cross-val\n42 misclas2 = kfoldLoss(cvMdl2); % calc losses\n43 cvMdl3 = crossval(mdl3, 'KFold', 4); % Guassian 4-fold cross-val\n44 misclas3 = kfoldLoss(cvMdl3); % calc losses
```

Code 3: q3.m

Question 4

Solution.

```
1 %{\n2 =====\n3 Homework 3 Question 4 Code\n4 Name      : Chase Lotito\n5 University : Southern Illinois University\n6 Course    : ECE469\n7 =====\n8 Description:\n9 K-Means Clustering\n10 =====\n11 %}\n12\n13 load fisheriris\n14 X = meas(:,3:4); % extract petal lengths and widths\n15\n16 % (A) X IS 2D, VISUALIZE IN 2D SPACE\n17 figure;\n18 plot(X(:,1), X(:,2), 'k*', 'MarkerSize', 5);\n19 title('Fisher''s Iris Data (Petal)');\n20 xlabel('Petal Lengths (cm)');\n21 ylabel('Petal Widths (cm)');\n22\n23 % (B) RUN K=3 K-MEANS\n24 rng(1); % reproducibility\n25\n26 % idx: vector of predicted cluster id's\n27 % C: matrix of centroid locations\n28 [idx, C] = kmeans(X,3);\n29\n30 % (C) COMPUTE CENTROID DISTANCE BY PASSING C INTO KMEANS\n31 % create grid\n32 x1 = min(X(:,1)):0.01:max(X(:,1));\n33 x2 = min(X(:,2)):0.01:max(X(:,2));\n34 [x1G,x2G] = meshgrid(x1,x2);\n35 XGrid = [x1G(:), x2G(:)];\n36 idx2Region = kmeans(XGrid, 3, MaxIter=1, Start=C);\n37\n38 % (D) VISUALIZE CLUSTERS\n39 figure;\n40 gscatter(XGrid(:,1), XGrid(:,2), idx2Region,\n41         [0,0.75,0.75;0.75,0,0.75;0.75,0.75,0], '.'); % vector is colors\n42 hold on;\n43 plot(X(:,1), X(:,2), 'k*', 'MarkerSize', 5);\n44 title('Fisher''s Iris Data (Petal)');\n45 xlabel('Petal Lengths (cm)');\n46 ylabel('Petal Widths (cm)');\n47 legend('Region 1', 'Region 2', 'Region 3', 'Data', Location='SouthEast');\n48 hold off;\n49\n50 % (E) EXTRACT LENGTH AND WIDTH OF SEPALS AND REPEAT ABOVE
```

```

50 X = meas(:,1:2); % extract sepal lengths and widths
51
52 % idx: vector of predicted cluster id's
53 % C: matrix of centroid locations
54 [idx, C] = kmeans(X,3);
55
56 % (C) COMPUTE CENTROID DISTANCE BY PASSING C INTO KMEANS
57 % create grid
58 x1 = min(X(:,1)):0.01:max(X(:,1));
59 x2 = min(X(:,2)):0.01:max(X(:,2));
60 [x1G,x2G] = meshgrid(x1,x2);
61 XGrid = [x1G(:), x2G(:)];
62 idx2Region = kmeans(XGrid, 3, MaxIter=1,Start=C);
63
64 % (D) VISUALIZE CLUSTERS
65 figure;
66 gscatter(XGrid(:,1), XGrid(:,2), idx2Region,
67         [0,0.75,0.75;0.75,0,0.75;0.75,0.75,0], '.'); % vector is colors
68 hold on;
69 plot(X(:,1), X(:,2), 'k*', 'MarkerSize',5);
70 title('Fisher''s Iris Data (Sepal)');
71 xlabel('Sepal Lengths (cm)');
72 ylabel('Sepal Widths (cm)');
73 legend('Region 1', 'Region 2', 'Region 3', 'Data', Location='SouthEast');
74 hold off;

```

Code 4: q4.m

Question 5

Solution.

```
1 %{\n2 =====\n3 Homework 3 Question 5 Code\n4 Name      : Chase Lotito\n5 University : Southern Illinois University\n6 Course    : ECE469\n7 =====\n8 Description:\n9 Principal Component Analysis (PCA)\n10 =====\n11 %}\n12\n13 % load hald (cement heat) dataset\n14 load('hald.mat');\n15\n16 % (i) PCA X, X ZERO MEAN (SAMPLE MEAN),\n17 %     COVARIANCE MATRIX, EIGENVECTORS\n18 %     OF COVARIANCE MATRIX\n19\n20 % let ingredients be X (13x4)-matrix\n21 X = ingredients;\n22\n23 % get zero mean for X (sample mean)\n24 meanX = mean(X(:));\n25 X = X - meanX;\n26 zero_meanX = mean(X(:));\n27 disp(['meanX = ', num2str(meanX), ' | zero_meanX = ', num2str(zero_meanX)\n28     ]);\n29\n30 % (4x4) covariance matrix (4 b/c 4 features)\n31 covX = cov(X);\n32\n33 % find the eigenvectors of covariance matrix\n34 % V: eigenvector column matrix\n35 % D: eigenvalue diagonal matrix\n36 [V,D] = eig(covX);\n37\n38 % sort the eigenvalues largest to smallest (noting location)\n39 [D_sort, idx] = sort(diag(D), 'descend');\n40 V_sort = V(:, idx); % rearrange eigenvector matrix accordingly\n41\n42 % (ii) use built-in MATLAB PCA function\n43 Mat_pc = pca(ingredients);\n44\n45 disp('MANUAL:');\n46 disp(V_sort);\n47 disp('BUILT-IN:');\n48 disp(Mat_pc);
```

Code 5: q5.m

APPENDIX A: Auxillary Code

The following codes was used to implement the Gaussian Kernel for Support Vector Machines (SVM).

```
1 function G = mysigmoid(U,V)
2 gamma = 1;
3 c = -1;
4 G = tanh(gamma*U*V' + c);
5 end
```

Code 6: mysigmoid.m

```
1 function G = mysigmoid_2(U,V)
2 gamma = 0.5;
3 c = -1;
4 G = tanh(gamma*U*V' + c);
5 end
```

Code 7: mysigmoid_2.m