ALDA Homework 3 – 2018 :

Submitted by :

Srinivasan Balan Unity id : sbalan

Harika Malapaka – Unity Id : hsmalapa

1. (12 points) [D-Separation][Ruth Okoilu] Conditional independence is a key concept in Bayesian belief network. Please answer the following conditional independence and d-separation questions using the graphs below.

(a) (3 points) In Figure 1 (left), are B and D d-separated given {A}? Justify your answer

X1={B}

X2={A}

X3={D}

The possible undirected paths from B to D are :

B→C→G→D (here sequential connection is there for nodes C and G, but they are not in X2 - so not blocked)

B→C→F→D (here sequential connection is there for nodes C and F, but they are not in X2 - so not blocked)

B→A←E→C→G→D (here A is convergent, but since it's in X2-it's not blocked)

B→A←E→C→F→D (here A is convergent, but since it's in X2-it's not blocked)

For the both paths above, the sequential connection is there for nodes C, G, F but as they are not in X2 – they are not blocked.

B→A←E→H→F→D (here A is convergent, but since it's in X2-it's not blocked)

B→C←E→H→F→D (here C is convergent, but since it's in X2-it's not blocked)

Serial connections at H and F, but as they are not in X2, they are not blocked.

No path has anything which is blocked by X2.

So not blocked.

Therefore B and D are not D-separated given A.

(b) (3 points) In Figure 1 (left), are A and D d-separated given {C, H}? Justify your answer.

The possible undirected paths from A to D are :

A←B→C→G→D

C is in X2 and has sequential connection – So Blocked.

A←E→C→G→D

C is in X2 and has sequential connection – So Blocked.

A←E→C→F→D

C is in X2 and has sequential connection – So Blocked.

A←E→H→F→D

H is in X2 and has sequential connection – So Blocked.

A←E→H→F→C→G→D

C is in X2 and has sequential connection – So Blocked.

A←B→C→F→D

C is in X2 and has sequential connection – So Blocked.

Since every possible undirected path is blocked – A and D are D-separated given {C,H}

(c) (3 points) In Figure 1 (right), are A and B d-separated given {F, E}? Justify your answer

The possible undirected paths between A and B are :

A→E→G→F→B

E is in X2 and has sequential connection – So Blocked.

A→E←H→F→B

F is in X2 and has sequential connection – So Blocked.

Since all the possible paths are blocked – A and B are D-separated given {F,E}

(d) (3 points) In Figure 1 (right), are C and D d-separated given {B}? Justify your answer.

X1={C}

X2={B}

X3={D}

The possible paths from C to D are :

C←E→H→F←D

C←E→G→F←D

F is converging.

It's descendants – B are in X2 : B.

it's not satisfying condition 3.

So not blocked.

Therefore, C and D are not D-separated given {B}

2. (15 points) [BN Inference][Song Ju] Compute the following probabilities according to the Bayesian net shown in Figure 2.

The joint distribution of the network is :

$P(A,B,C,D,E)=P(A) * P(B|A)*P(C|A)*P(E|B)*P(D|B,C)$

(a) (5 points) Compute P(E). Show your work.

$P(E)=\sum(A,B,C,D)\ [P(A) * P(B|A)*P(C|A)*P(E|B)*P(D|B,C)]$

We can normalize C and D as they will be equal to 1 when we take all possible combinations.

Then we will be left with ∑(A,B) [P(A) * P(B|A) * P(E|B)]

We have 4 cases :

P(A) * P(B|A) * P(E|B) +

P(A) * P(-B|A) * P(E|-B) +

P(-A) * P(B|-A) * P(E|B) +

P(-A) * P(-B|-A) * P(E|-B)

= 0.75 * 0.2 *0.6 +

0.75 * 0.8 * 0.3 +

0.25 * 0.5 * 0.6 +

0.25 * 0.5 * 0.3

= 0.09+ 0.18 + 0.075 + 0.0375

= 0.3825

(b) (5 points) Compute P(~ B,C,D,E). Show your work.

P(~ B,C,D,E)=∑(A) [P(A) * P(-B|A)*P(C|A)*P(E|-B)*P(D|-B,C)]

If A=t:

0.75 * 0.8 * 0.7 * 0.3 * 0.1

=0.0126

If A=f :

0.25 * 0.5 * 0.25 * 0.3 * 0.1

=0.00093

0.0.126+0.00093=0.0135

P(~ B,C,D,E)=0.0135

(c) (5 points) Compute P(D | A). Show your work.

Checking for D-Separation:

X1={D}

X2={}

X3={A}

The possible undirected paths are :

D← B← A

D←C← A

Although serial connection at B and C – they are not in Evidence – Not Blocked.

They are not D-separated – So needs to be calculated.

P(D|A) = ∑(B,C,E) [P(A) * P(B|A)*P(C|A)*P(E|B)*P(D|B,C)] /

∑(B,C,D,E) [P(A) * P(B|A)*P(C|A)*P(E|B)*P(D|B,C)]

Normalizing few terms, we get :

P(D|A) = ∑(B,C) [P(A) * P(B|A)*P(C|A)*P(D|B,C)] /

∑(B,C) [P(A) * P(B|A)*P(C|A)]

We can cancel P(A) in both numerator and denominator.

The denominator is now 1 because we can marginalize B and c as well.

[P(B|A) * P(C|A) * P(D|B,C) +

P(B|A) * P(-C|A) * P(D|B,-C) +

P(-B|A) * P(C|A) * P(D|-B,C) +

P(-B,A) * P(-C|A) * P(D|-B,-C)]

= [0.2 * 0.7 * 0.3 +

0.2 * 0.3 * 0.25 +

0.8 * 0.7 * 0.1 +

0.8 * 0.3 * 0.35]

=[0.042 + 0.015 + 0.056+0.084]

=0.197

Therefore , P(D|A) = 0.197

3. (20 points) [LR][Xi Yang]

(a) (7 points) Given the following three data points of (x, y): (1,2), (2,1), (0,−1), try to use a linear regression y = β1x+β0 to predict y. Determine the values of β1 and β0 and show each step of your work.

y = β1x+β0

To reduce the error, we should minimize the argmin∑(y-(β1x+β0))^2

Now, to get β1 and β0, we differentiate the above equation once with respect to β1 and β0 and equate them to 0.

$$\frac{\partial}{\partial \beta_0} \Sigma \left[y - (\beta_1 x + \beta_0) \right]^2 = 0$$

$$\frac{\partial}{\partial} = \Sigma \left\{ 2(y - (\beta_1 x + \beta_0)) \right\} = 0$$

Dividing both sides by 2, we get

$$\Sigma \left[y - \beta_1 x - \beta_0 \right] = 0$$

$$\Rightarrow \Sigma y - \Sigma \beta_1 x - \Sigma \beta_0 = 0$$

$$\Sigma \beta_0 = \Sigma y - \Sigma \beta_1 x$$

$$\Rightarrow \beta_0 = \frac{\Sigma y - \Sigma \beta_1 x}{n}$$

$$\frac{\partial}{\partial \beta_1} \Sigma \left[y - (\beta_1 x + \beta_0) \right]^2 = 0$$

$$= \Sigma \left\{ 2(y - (\beta_1 x + \beta_0))(-x) \right\} = 0$$

taking -2 common, & dividing by both sides.

$$\Sigma (yx) - \Sigma \beta_1 x^2 - \Sigma \beta_0 x = 0$$

$$\Sigma (yx) - \Sigma \beta_0 x = \Sigma \beta_1 x^2$$

$$\Rightarrow \beta_1 = \frac{\Sigma yx - \beta_0 \Sigma x}{\Sigma x^2}.$$

Using the values β_0 & β_1,

$$\beta_0 = \frac{\sum y_i - \beta_1 x_i}{n} \qquad \beta_1 = \frac{\sum x_i y_i - \sum x_i w_0}{\sum x_i^2}$$

$$\beta_0 = \frac{[2 - \beta_1(1)] + [1 - \beta_1(2)] + [-1 - \beta_1(0)]}{3}$$

$$\beta_0 = (2 - \beta_1 + 1 - 2\beta_1 - 1)/3$$

$$\beta_0 = \frac{2 - 3\beta_1}{3}$$

$$\beta_1 = \frac{[(1)(2) - 1\beta_0] + [(2)(1) - 2\beta_0] + [0]}{(1)^2 + (2)^2 + 0}$$

$$\beta_1 = \frac{2 - \beta_0 + 2 - 2\beta_0}{5} \quad | \Rightarrow 5\beta_1 = 4 - 2 + 3\beta_1$$

$$\beta_1 = \frac{4 - 3\beta_0}{5} \quad | \Rightarrow 2\beta_1 = 2$$

$$| \Rightarrow \beta_1 = 1$$

$$\beta_1 = \frac{4 - 3\left(\frac{2 - 3\beta_1}{3}\right)}{5} \quad | \quad \beta_0 = \frac{2 - 3(1)}{3}$$

$$| \quad \beta_0 = -1/3.$$

Therefore β0 = -1/3 and β1=1

(b) (13 points) [Programming Task] Apply the following three linear regressions:
(1) y = α1x1 + α2x2 + α3x3 + α4x4 + α0

(2) y = β1x2 1 + β2x2 2 + β3x2 3 + β4x2 4 + β0

(3) y = γ1x3 1 + γ2x3 2 + γ3x3 3 + γ4x3 4 + γ0

to the provided data file "hw3q3(b).csv", which is from a combined cycle power plant dataset (https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+ Plant). In the given data file, xi,i ∈ [1,4] are four features and y is the prediction target which indicates hourly electrical energy output.

Write code in Matlab, R or Python to perform following tasks. Please report your outputs and key codes in the document file and also include your code (end with .m, .r or .py) in the .zip file.

i. (6 points) Load the data. Fit the whole dataset to the three linear regression models, respectively. Report the coefficients (αs, βs, γs) of the three models.

model 1

Intercept 500.207082

x1 -12.389265

x2 2.800598

x3 -12.327601

x4 -64.679164

dtype: float64

model 2

Intercept 477.097890

np.power(x1, 2) -6.055810

np.power(x2, 2) 7.284263

np.power(x3, 2) -15.383582

np.power(x4, 2) -54.342367

dtype: float64

model 3

Intercept 466.283686

np.power(x1, 3) 1.248777

np.power(x2, 3) 16.380038

np.power(x3, 3) -24.203288

np.power(x4, 3) -43.633282

dtype: float64

ii. (7 points) Use leave-one-out cross validation to determine the RMSE (root mean square error) for the three models. Specifically, in each fold, fit the training data to the model to determine the coefficients, then apply the coefficients to get predicted label for testing data (You don't need to report the coefficients in each fold). Report RMSE for the three models. Based on the RMSE, which model is the best for fitting the given data?

Refer the code 3b-linReg.py for clarifications.

The RMSE values are (Using Pandas formula for square root):

3.6503620104287067

5.209585068655514

6.56548306522179

The RMSE values are (using Numpy formula for square root):

4.492722863492924

6.458732074830128

8.086443688167645

In any case, Model 1 would be best as it's RMSE value is low indicating that difference between predicted and actual values is actually low.

There, for the given data, Model 1 is best.

4. (16 points) (extra 5 points) [ANN] [Ruth Okoilu] Train, validate, and test a neural network model using the dataset in hw3q4.zip, which contains training data (75%), validation data (12.5%), and test data (12.5%). There are two output classes in this data set. You can either choose matlab or a python neural networks package, Keras for this problem. (All the output should be included in your report. Otherwise, your points are deducted.)

(a) (5 extra points only for choosing Keras) Please briefly describe how to construct your working environments (e.g. language, package version, backend for neural networks, installation, etc.) in your report, and write how to execute your codes on 'readme' file.

- Keras Package was choosen for implementing the Neural Net.
- Linus OS – Ubuntu was used for this task.
- Keras uses tensor Flow or Theano as Backend
- We had used TensorFlow.
- The language used is Python
- Version in Python is 3.5.2
- To execute this, we should have Python3 installed in the system.
- On Ubuntu, it's available once Ubuntu 64 bit is installed.
- The file nn.py has the code for question 4
- Before executing , make sure these packages are already there: keras and tensorFlow.
- Else we can install using :
- apt install python3-pip
- pip3 is the the package management system for Python.
- 3 is since we are using Python3.
- Once pip3 is installed, the following 2 can be installed easily.
- pip3 install tensor-flow
- pip3 install keras
- pip3 install pandas
- pip3 install numpy

NOTE : sometimes, due to permission issues, we may use ‘sudo’ before every command in Ubuntu.

(b) (8 points)

(1) Construct neural networks using the given training dataset (X train, Y train) using different number of hidden neurons. Set the parameters as follows: activation function for hidden layer=’relu’, activation for output layer =’sigmoid’, loss function =’mse’, metrics= ’accuracy’, epochs=10, batch size=50. For each model, change the number of hidden neurons in the order of 2, 4, 6, 8, 10.

Kindly use the code nn.py for any information

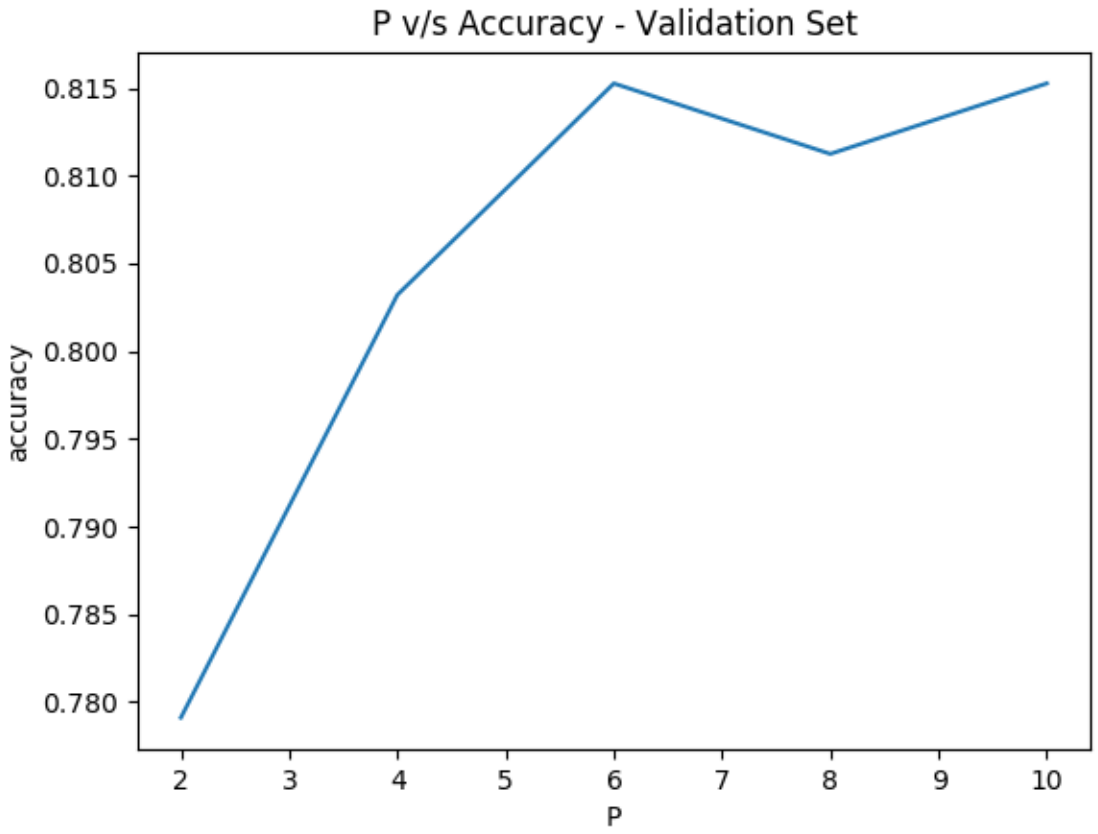
(2) Validate each neural network using the given validation dataset (X val, Y val). The validation accuracy is used to determine how many number of hidden neurons are optimal for this problem. Provide the core code for ”neural network learning” with comments in your report. (Please apply a fixed random seed 7 in order to generate a same result every time.)

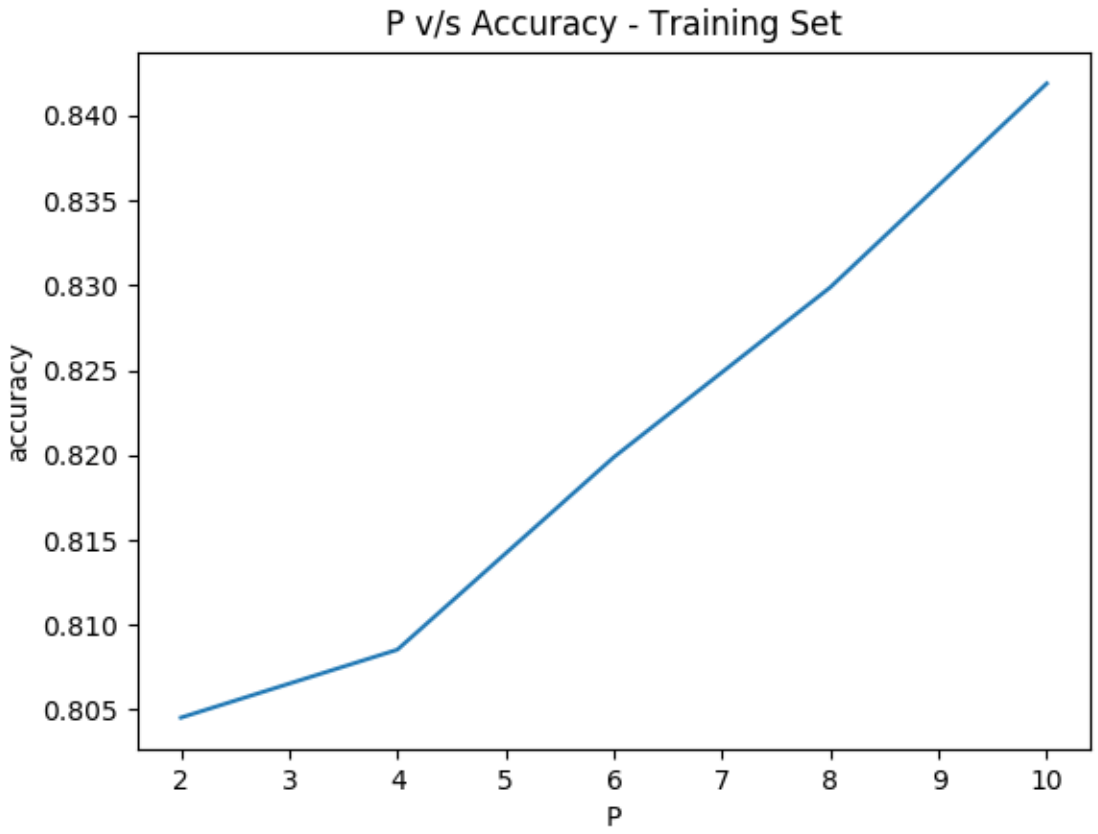
Model Snippet (Core Code – with comments)

```
#Add a hidden dense layer with input dimensions as 61, since we have 61 attributes.

# Add i output neurons where i is in (2,4,6,8,10) and activation function as relu

classifier.add(Dense(output_dim = i, init = 'uniform', activation = 'relu', input_dim = 61))

# add output layer with activation function as sigmoid

# output dimension is 1 since we want only 1 output node for classification problem in neural net

classifier.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))

# compile the model
```

```
classifier.compile(optimizer = 'adam', loss = 'mse', metrics = ['accuracy'])

# fit the training and test data according with batch and epoch sizes

classifier.fit(X_train,Y_train, batch_size = 50, epochs = 10)

pred_train=np.round(classifier.predict(X_train))

pred_val=np.round(classifier.predict(X_val))
```

(c) (3 points) Plot a figure, where the horizontal x-axis is the number of hidden neurons, and the vertical y-axis is the accuracy. Please plot both training and validation accuracy in your figure. (Note that the exact accuracy could be slightly different according to your working environments, however you can analyze the trend.)


P v/s Accuracy - Validation Set
accuracy
0.815
0.810
0.805
0.800
0.795
0.790
0.785
0.780
2
3
4
5
6
7
8
9
10
P



(d) (3 points) Provide a simple analysis about your results and choose the optimal number of hidden neuron from the analysis.

Analysis :

As per the graph, we can see that as we keep increasing the number of hidden neurons, it's not that the accuracy also increases. Because in validation set, there is a dip at p=8.

There would be a threshold and after that, it doesn't increase so much.

For Training Set, 10 neurons gave the best accuracy.

And then, while increasing the neurons, the accuracy increases.

For Validation set, the trend is almost same, except for it reached a threshold 6 and then started to decrease and again for 10 neurons, it increased.

Considering Validation set, we would chose the optimal number of neurons to be 6 or 10.

Overfitting might happen as we increase neurons because the model may get overfitted as complexity is increasing.

(e) (2 points) Report the test accuracy using the given test dataset (X test, Y test) on the neural network with the optimal number of hidden neurons.

The best accuracy for Validation set appeared when p is set as 6 and 10.

So using p=10 number of neurons for test set, the accuracy of test is 0.79 (when hidden neurons=4)

Although for p=6 in validation set, the accuracy is same as p=10, the accuracy for testing set is more for p=10.

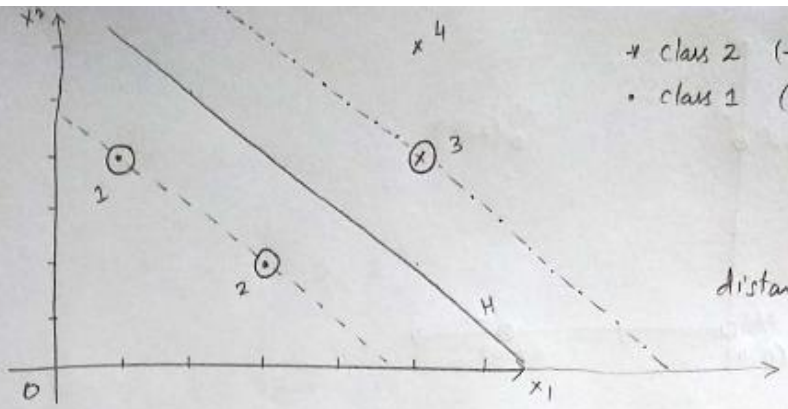
5. (22 points) [SVM Theory]

(a) (10 points) [Song Ju] Support vector machines (SVM) learn a decision boundary leading to the largest margin between classes. In this question, you'll train a SVM on a tiny dataset with 4 data points as shown in Figure 3. This dataset consists of two points with class1 (label 1) and two points with class2 (label −1).

i. (5 points) Find the weight vector w and bias b. What is the equation corresponding to the decision boundary?

Problem: 5

a.



distance from H to (+) = $\sqrt{2}$
H to (-) = $\sqrt{2}$

i) distance between points 2 & 3 : $2\sqrt{2}$ (Euclidean when H passes (4,3) & (3,4)

distance between points 2 & 3 : 2 (vertical line $x = 4$ as H

Therefore the possible support vector are passing through (3,2), (1,4) in class (+1) and 5,4 in class (-1)

$$w^T x_i + w_0 = +1 \quad \text{class (+)}$$

$$w^T x_i + w_0 = -1 \quad \text{class (-)}$$

$$3w_1 + 2w_2 + w_0 = 1$$

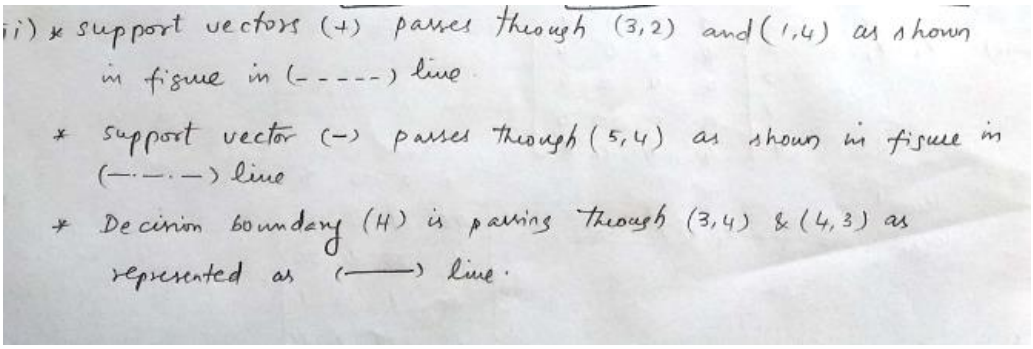
$$w_1 + 4w_2 + w_0 = 1$$

$$5w_1 + 4w_2 + w_0 = -1$$

solving w_1, w_2, w_0, we get $w_1 = w_2 = -1/2$ $\quad w^T = \left[\frac{-1}{2} \quad \frac{-1}{2}\right]$

decision boundary is $-\frac{1}{2}x_1 - \frac{1}{2}x_2 + \frac{7}{2} = 0$ or $x_1 + x_2 = 7$ $\quad w_0 = 7/2$ $\quad \therefore b = 7/2$

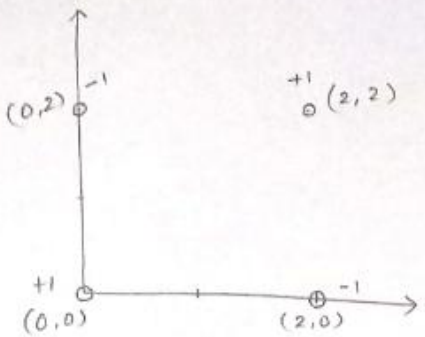
ii. (5 points) Circle the support vectors and draw the decision boundary.



(b) (12 points) [Xi Yang] Given 2-dimensional data points Xi,i ∈ [1,2,3,4] as shown in Table 1, in this question, you will employ the kernel function for SVM to classify these four data points.

i. (4 points) Suppose the kernel function is: K(Xi,Xj) = (1 + Xi ·Xj)2, where Xi and Xj indicate two data points. This kernel is equal to an inner product ϕ(Xi)·ϕ(Xj) with a certain function of ϕ. What is the function of ϕ?

b)



	1	2	3	4
x_i	(0,2)	(2,0)	(0,0)	(2,2)
y_i	-1	-1	+1	+1

i) $K(x_i, x_j) = (1 + x_i \cdot x_j)^2$ (polynomial kernel)

$x \longrightarrow \phi(x)$

$\phi(x) = \left[x_1^2,\ x_2^2,\ \sqrt{2}x_1x_2,\ \sqrt{2}x_1,\ \sqrt{2}x_2,\ 1 \right]^T$

$\phi(x') = \left[x_1'^2,\ x_2'^2,\ \sqrt{2}x_1'x_2',\ \sqrt{2}x_1',\ \sqrt{2}x_2',\ 1 \right]^T$

$K(x', x) = \phi(x')^T \cdot \phi(x)$

$= x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1x_1'x_2x_2' + 2x_1x_1' + 2x_2x_2' + 1$

$= (x_1x_1' + x_2x_2' + 1)^2$

$= (1 + x^T x')^2$

ii. (2 points) Transform the four given data points Xi,i ∈ [1,2,3,4] to the higher dimensional space via the function ϕ get from

ii) $x_i \cdot x_j'$ =

	1	2	3	4
1	4	0	0	4
2	0	4	0	4
3	0	0	0	0
4	4	4	0	8

Kernel

$K(x_i \cdot x_i') = (1 + x_i \cdot x_i')^2$ =

	1	2	3	4
1	25	1	1	25
2	1	25	1	25
3	1	1	1	1
4	25	25	1	81

$\phi(x^1) = [0, 4, 0, 0, 2\sqrt{2}, 1]$

$\phi(x^2) = [4, 0, 0, 2\sqrt{2}, 0, 1]$

$\phi(x^3) = [0, 0, 0, 0, 0, 1]$

$\phi(x^4) = [4, 4, 4\sqrt{2}, 2\sqrt{2}, 2\sqrt{2}, 1]$

iii. (6 points) Assume the four transformed data points get from (ii) are all support vectors. Apply Lagrange multipliers to determine the maximum margin linear decision boundary in the transformed higher dimensional space.

iii) $L = \sum_{i}^{4} \alpha_i - \frac{1}{2} \sum_{i} \sum_{j} \alpha_i \alpha_j y_i y_j k(x_i \cdot x_j)$

$= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2} [25\alpha_1^2 + \alpha_1\alpha_2 - \alpha_1\alpha_3 - 25\alpha_1\alpha_4$
$\alpha_1\alpha_2 + 25\alpha_2^2 - \alpha_3\alpha_2 - 25\alpha_2\alpha_4$
$-\alpha_1\alpha_3 - \alpha_2\alpha_3 + \alpha_3^2 + \alpha_3\alpha_4$
$-25\alpha_1\alpha_4 - 25\alpha_2\alpha_4 + \alpha_3\alpha_4 + 81\alpha_4^2]$

$\frac{\partial L}{\partial \alpha_1} = 0 \Rightarrow \quad 1 - \frac{1}{2}[2 \cdot 25\alpha_1 + \alpha_2 - \alpha_3 - 25\alpha_4 + \alpha_2 - \alpha_3 - 25\alpha_4] = 0$

$$25\alpha_1 + \alpha_2 - \alpha_3 - 25\alpha_4 = 1 \quad (A)$$

$\frac{\partial L}{\partial \alpha_2} = 0 \Rightarrow \quad 1 - \frac{1}{2}[2 \cdot 25\alpha_2 + \alpha_1 + \alpha_1 - \alpha_3 - \alpha_3 - 25\alpha_4 - 25\alpha_4] = 0$

$$25\alpha_2 + \alpha_1 - \alpha_3 - 25\alpha_4 = 1 \quad (B)$$

$\frac{\partial L}{\partial \alpha_3} = 0 \Rightarrow \quad 1 - \frac{1}{2}[-\alpha_1 - \alpha_2 - \alpha_1 - \alpha_2 + 2\alpha_3 + \alpha_4 + \alpha_4] = 0$

$$-\alpha_1 - \alpha_2 + \alpha_3 + \alpha_4 = 1 \quad (C)$$

$\frac{\partial L}{\partial \alpha_4} = 0 \Rightarrow \quad 1 - \frac{1}{2}[-25\alpha_1 - 25\alpha_1 - 25\alpha_2 - 25\alpha_2 + \alpha_3 + \alpha_3 + 2 \cdot 81 \cdot \alpha_4] = 0$

$$-25\alpha_1 - 25\alpha_2 + \alpha_3 + 81\alpha_4 = 1 \quad (D)$$

solving A, B, C, D, we get $\alpha_1 = 0.2083$ $\alpha_2 = 0.2083$ $\alpha_3 = 1.2916$ $\alpha_4 = 0.1$

$\alpha_1 = 5/24$ $\alpha_2 = 5/24$ $\alpha_3 = 31/24$ $\alpha_4 = 1/8$

$W = \sum_{i \in SV} \alpha_i y_i \phi(x_i)$ Let $\phi(x) < x_1' \; x_2' \; x_3' \; x_4' \; x_5' \; x_6'$ new 6 dimension vector

decision boundry $\boxed{W X + w_0 = +1}$ for 1 sv whose class (+1)

$\Rightarrow \sum_{i \in SV} \alpha_i y_i \phi(x_i) . \phi(x) + w_0$ (from equation ①) $= 0$

$\Rightarrow \frac{-5}{24} \langle 0, 4, 0, 0, 2\sqrt{2}, 1 \rangle \cdot \langle x_1' \; x_2' \; x_3' \; x_4' \; x_5' \; x_6' \rangle$

$\frac{-5}{24} \langle 4, 0, 0, 2\sqrt{2}, 0, 1 \rangle \cdot \langle x_1' \; x_2' \; x_3' \; x_4' \; x_5' \; x_6' \rangle$

$\frac{31}{24} \langle 0, 0, 0, 0, 0, 1 \rangle \cdot \langle x_1' \; x_2' \; x_3' \; x_4' \; x_5' \; x_6' \rangle$

$\frac{1}{8} \langle 4, 4, 4\sqrt{2}, 2\sqrt{2}, 2\sqrt{2}, 1 \rangle \cdot \langle x_1' \; x_2' \; x_3' \; x_4' \; x_5' \; x_6' \rangle = 0$

$\Rightarrow \frac{-5}{24}(4x_1') + \frac{1}{8}(4x_1') \quad \frac{-5}{24}(4x_2') + \frac{1}{8}(4x_2') + \frac{1}{8} 4\sqrt{2} x_3'$

$\frac{-5}{24} 2\sqrt{2} x_4' + \frac{1}{8} 2\sqrt{2} x_4' - \frac{5}{24}(2\sqrt{2}) x_5' + \frac{2\sqrt{2} x_5'}{8} - \frac{5}{24}(x_6' + x_6') + \frac{31}{24} x_6' + \frac{1}{8} x_6$

$= 0$

$\Rightarrow -\frac{8x_1'}{24} - \frac{8}{24} x_2' + \frac{1}{8} 4\sqrt{2} x_3' - \frac{4\sqrt{2} x_4'}{24} - \frac{4\sqrt{2} x_5'}{24} + x_6' = 0$

$\Rightarrow -\frac{x_1'}{3} - \frac{x_2'}{3} + \frac{1}{\sqrt{2}} x_3' - \frac{1}{3\sqrt{2}} x_4' - \frac{1}{3\sqrt{2}} x_5' + x_6' = 0$

$\Rightarrow \langle \frac{-1}{3} \quad \frac{-1}{3} \quad \frac{1}{\sqrt{2}} \quad \frac{-1}{3\sqrt{2}} \quad \frac{-1}{3\sqrt{2}} \quad 1 \rangle \cdot \langle x_1' \; x_2' \; x_3' \; x_4' \; x_5' \; x_6' \rangle$

$= 0$

The decision boundary is

$$\boxed{-\frac{x_1'}{3} - \frac{x_2'}{3} + \frac{x_3'}{\sqrt{2}} - \frac{x_4'}{3\sqrt{2}} - \frac{x_5'}{3\sqrt{2}} + x_6' = 0}$$

Maximum Margin, $w = \sum_{i \in SV} \alpha_i y_i \phi(x_i)$

$$[w_1\ w_2\ w_3\ w_4\ w_5\ w_6] = \alpha_1 y_1 \phi(x^1) + \alpha_2 y_2 \phi(x^2) + \alpha_3 y_3 \phi(x^3) + \alpha_4 y_4 \phi(x^4)$$

$$= \frac{-5}{24}[0, 4, 0, 0, 2\sqrt{2}, 1] + \frac{-5}{24}[4, 0, 0, 2\sqrt{2}, 0, 1] + \frac{31}{24}[0, 0, 0, 0, 0, 1] + \frac{1}{8}[4, 4, 4\sqrt{2}, 2\sqrt{2}, 2\sqrt{2}, 1]$$

$$[w_1\ w_2\ w_3\ w_4\ w_5\ w_6] = \left[-\tfrac{1}{3}, \tfrac{-1}{3}, \tfrac{1}{\sqrt{2}}, \tfrac{-1}{3\sqrt{2}}, \tfrac{-1}{3\sqrt{2}}, 1\right]$$

$$\text{magnitude of } w = \sqrt{\left(\tfrac{-1}{3}\right)^2 + \left(\tfrac{-1}{3}\right)^2 + \left(\tfrac{1}{\sqrt{2}}\right)^2 + \left(\tfrac{-1}{3\sqrt{2}}\right)^2 + \left(\tfrac{-1}{3\sqrt{2}}\right)^2 + 1^2}$$

$$= \sqrt{\frac{11}{6}}$$

$$\text{margin} = \frac{2}{\|w\|} = \frac{2}{\sqrt{11/6}}$$

$$= 1.477$$

$w = \sum_{i \in SV} \alpha_i y_i \phi(x_i)$

Taking 1st SV with class $y_i = -1$

$w x_1 + w_0 = -1$

$w \phi x_1 + w_0 = -1$

$\sum_{i \in SV} \alpha_i y_i \phi(x_i) \cdot \phi(x_1) + w_0 = -1$

$$-\frac{5}{24}(25) \quad -\frac{5}{24}(1) \quad +\frac{31}{24}(1) + \frac{1}{8}(25) + w_0 = -1$$

$$w_0 = \frac{125}{24} + \frac{5}{24} - 1 - \frac{31}{24} - \frac{25}{8}$$

$$= \frac{125 + 5 - 31 - 24 - 75}{24}$$

$$\boxed{w_0 = 0} \quad \text{——} \quad (1)$$

If we consider margin as 2/|w|, it's then 1.477.
Else |w| = √11/6.

6. (15 points) [SVM Programming][Xi Yang] In this question, you will employ SVM to solve a classification problem for the provided data file "hw3q6.csv". Each row in the data file indicates a sample. The first 12 columns are features and the last column "Class" indicates the label, with 1 and 0 indicating the positive and negative samples, respectively. Write code in Matlab, R or Python to perform the following tasks. Please report your outputs and key codes in the document file and also include your code (end with .m, .r or .py) in the .zip file.

(a) (1 point) Load data. Report the size of positive and negative samples in dataset.

90 positive

110 negative

(b) (4 points) Use stratified random sampling to divide the dataset into training data (75%) and testing data (25%). Report the number of positive and negative samples in both training and testing data.

train set: positive 67

train set : negative 83

test set : positive 23

test set : negative 27

(c) (4 points) Take SVM with linear kernel as classifier (third-party packages are allowed to use) and set the regularization parameter C as: [0.1, 0.5, 1, 5, 10, 50, 100], respectively. For each value of C, train a SVM classifier with the training data and get the number of support vectors (SVs). Generate a plot with C as the

horizontal axis and number of SVs as the vertical axis. Give a brief analysis for the plot.

The support vectors in each class for given C values are respectively:

The value outside brackets are the total number of support vectors.

[50 50] 100

[28 24] 52

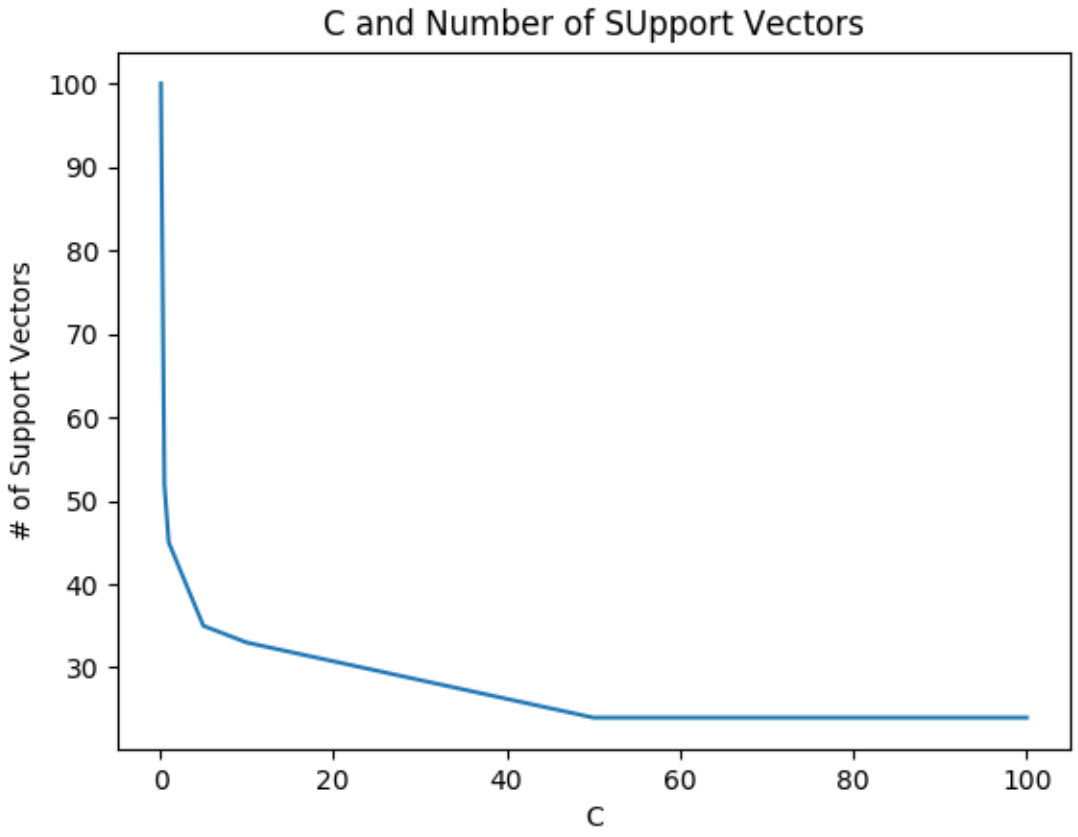
[23 22] 45

[19 16] 35

[21 12] 33

[14 10] 24

[14 10] 24



NOTE : these are the support vectors generated for random sampling with random=10.

It will vary as we vary the randomness

Analysis on Plot :

- As we could observe, the number of support vectors is decreasing as we increase the C value which is regularization parameter.
- As we increase the penalty for SVM model making mistakes, we could see that the number of support vectors it needs is decreasing.
- Apart from that, it reaches a threshold and then from there, it's constant and no longer decreasing.
- Example, here from the penalty 60 onwards, the support vectors required is not dropping/changing much.

(d) (6 points) Compare 4 different kernel functions, including linear, polynomial, radial basic function (Gaussian kernel), and sigmoid kernel. Make a table to

record the accuracy, precision, recall and f-measure of the classification results for the 4 kernel functions. Try to tune the parameters via grid search and report your best results with the optimal parameters. Based on the results, which kernel function will you choose?

First, grid search was applied on 4 kernal functions.

For each of them, here are the best parameters with accuracies .

The parameters chosen were :

parameters = {'C':[0.1,0.5,1,5,10,50,100],'gamma':[0.1,2,5,10],
'degree':[2,3,4,5],'coef0':[0,1,2.5, 4,5.5]}

Although few parameters are not applicable for few kernels, the grid search would simply ignore it.

So it was considered as a common set of parameters for all 4 functions.

The output of grid search for 4 kernels is (along with accuracy):

Linear

0.9133333333333333

SVC(C=10, cache_size=200, class_weight=None, coef0=0,

decision_function_shape='ovr', degree=2, gamma=0.1, kernel='linear',

max_iter=-1, probability=False, random_state=None, shrinking=True,

tol=0.001, verbose=False)

RBF

0.96

SVC(C=50, cache_size=200, class_weight=None, coef0=0,

decision_function_shape='ovr', degree=2, gamma=0.1, kernel='rbf',

max_iter=-1, probability=False, random_state=None, shrinking=True,

tol=0.001, verbose=False)

Poly

0.96

SVC(C=0.1, cache_size=200, class_weight=None, coef0=1,

decision_function_shape='ovr', degree=4, gamma=0.1, kernel='poly',

max_iter=-1, probability=False, random_state=None, shrinking=True,

tol=0.001, verbose=False)

Sigmoid

0.8333333333333334

SVC(C=0.1, cache_size=200, class_weight=None, coef0=0,

decision_function_shape='ovr', degree=2, gamma=0.1, kernel='sigmoid',

max_iter=-1, probability=False, random_state=None, shrinking=True,

tol=0.001, verbose=False)

The table for tor the 4 kernals with optimal parameters is :

	kernal	Accuracy	Precision	Recall	F-measure
1	linear	0.88	0.814815	0.956522	0.88
2	rbf	0.88	0.814815	0.956522	0.88
3	poly	0.96	0.92	1	0.958333
4	sigmoid	0.88	0.814815	0.956522	0.88

Based on the above results, Polynomial kernel is the best kernel for the given dataset.

It’s accuracy is high.(96%)

Even it’s recall is high among others (recall is 1).

The reason for chosing poly is, not only for this particular random seed and parameter setting, but for many settings, it gave better results.

Polynomial kernel even considers paramters of attributes, hence it’s performance might better than others, atleast in the above case.