Arda Cankat BATI

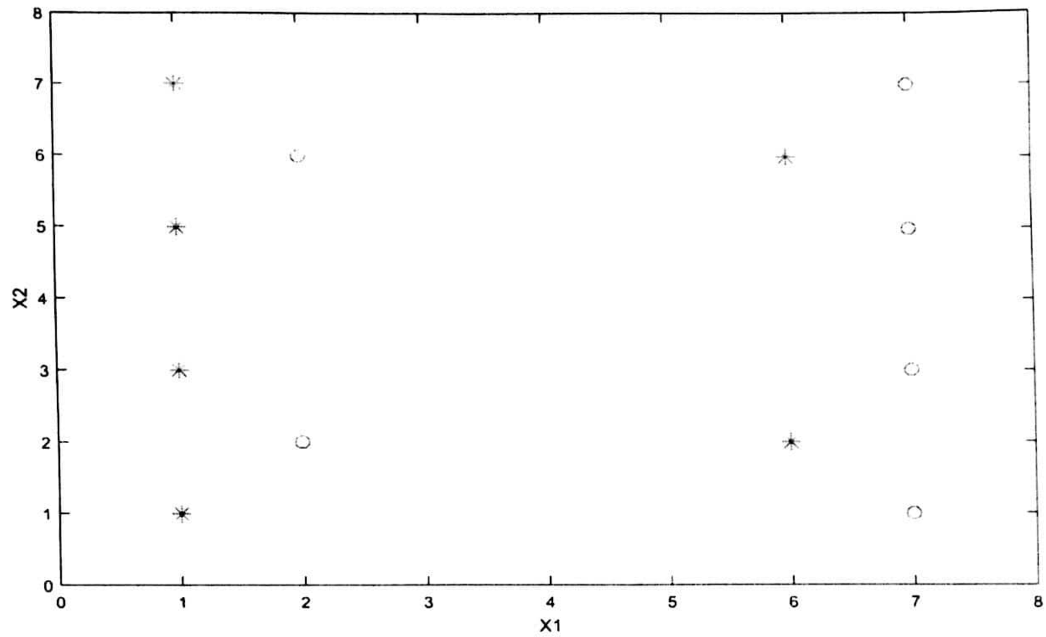
CS 464: Homework Assignment 2

Due: April 17 2017 17:00 pm

Instructions

- You will submit a hard copy for the answers of the write up questions (including the plots) and will upload the code online on Moodle by the due date. You may hand in the hard copy in classes to the instructor or may drop it in the box in room EA427 (please stick to this submission routes) and please STAPLE your write up.
- You may code in any programming language you would prefer (however, using Matlab is strongly recommended for this homework). In submitting the code on moodle, package your code as a ZIP file with the name convention explained in "How to Submit Programming Assignments" document available at Moodle.
- **If you are submitting the homework late (see the late submission policy in syllabus), prepare a soft copy for all the parts of the homework and submit it on Moodle. Moodle will allow late submissions until after 4 days of submission, but we will grade your homework based on the time stamp and your remaining late days.**
- Please refer to the syllabus for policies regarding collaboration, and extensions policy.
- <u>Important</u>: If you fail to follow the submission instructions, such as uploading your submission with a wrong name, you will receive **zero** for this assignment. There will be no exceptions, so please follow the instructions.

1 Decision Tree [15 pts]



Consider the dataset shown in Figure above. You will train a decision tree using information gain as the splitting criteria. Consider the following stopping criteria (ie. early pruning criteria): if the entropy of a node is below a predefined threshold T, stop splitting that node, and set it as a leaf.

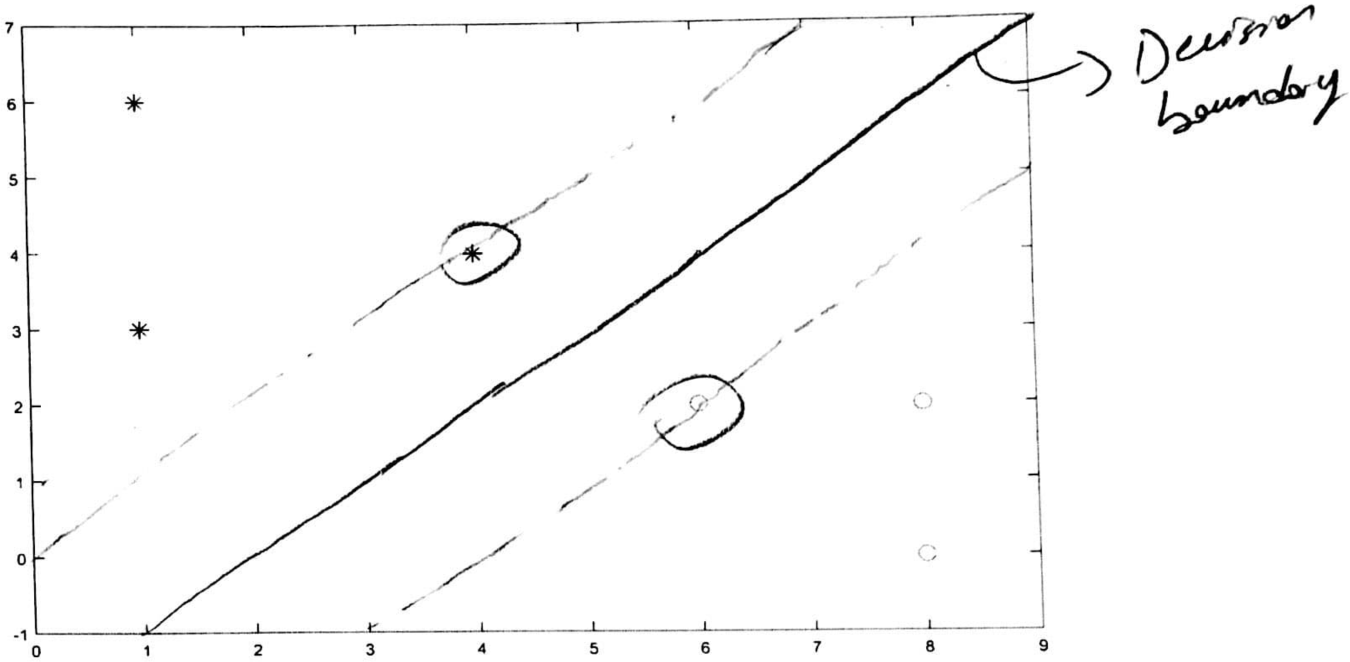
a) [5 pts] Draw the decision tree that will be drawn for this dataset without any pruning (T=0).

b) [5 pts] Draw the decision boundaries resulting from the decision tree you drawn for part a.

c) [5 pts] Assume we have a very large dataset, Draw a hypothetical plot, which show the training and test accuracies (Y-axis) as T changes from 1 to 0 (X-axis). Explain your rationale.

2 Support Vector Machines (SVM) [15 pts]

Consider the dataset below which consists of two classes each with 3 points. Stars represent negative class and circles represent positive class.



a) [3 pts] Draw the decision boundary and circle all support vectors.
b) [5 pts] Calculate the parameters ($\vec{w}$ and b) of the decision boundary $h(u) = \vec{w}.\vec{u} + b \geq 0$ for the SVM solution to part a.
c) [4 pts] Show the weight(alpha) of each point.
d) [2 pts] Suppose we moved the point at (4,4) to (5,3), how will the weight(alpha) of this point change? Just tell if it will increase or decrease.

3 Object recognition using SVM [40 pts]

In this part, you will separate handwritten B characters from P characters in UCI letter recognition dataset[1], using Support Vector Machines (SVM). Download the zip file `HW2data` on Moodle and you will find `Bs.csv` and `Ps.csv`. For Matlab you can also use `HW2data.mat`, which has Bs and Ps in it. Now, randomly divide the data for Bs and Ps into train and test (roughly 70% - 30% for each class, respectively) and save it because you will use the same data for the next part.

You might prefer to use LibSVM [2], Matlab or any other SVM package.

a) Linear Kernel [15 pts] Train an SVM classifier with linear kernel on your training set. You also need to fine-tune your classifier with cross-validation to find the best cost parameter (C). To do that, start from some small C value like 0.001 and step through larger values and perform cross-validation to measure the quality of each C value (Hint: You can try values like $10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}$). Plot cross-validation accuracy values over your tuning process. Report the highest cross validation accuracy, and the corresponding C value. Re-train a model using the best C value and run it on the test set. Output the decision values of test set as a file. Declare your test result.

[1]https://archive.ics.uci.edu/ml/datasets/Letter+Recognition
[2]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

b) Radial Basis Function Kernel [25 pts] Use SVM with RBF kernel. RBF kernel is

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||^2}{2\sigma^2}\right) \tag{1}$$

$||\mathbf{x}-\mathbf{x}'||^2$ is the squared Euclidean distance between the two feature vectors. σ is a free parameter. An equivalent, but simpler, definition involves a parameter $\gamma = -\frac{1}{2\sigma^2}$:

$$K(\mathbf{x}, \mathbf{x}') = \exp(\gamma||\mathbf{x} - \mathbf{x}'||^2) \tag{2}$$

Kernel SVM requires C and γ parameters to be tuned ($\gamma = \frac{-1}{2\sigma^2}$). To do that, keep a C value constant as you are trying some range of γ values. Update C again and apply same procedure to γ and iterate up to some end condition. You may use each C and γ pair, where $C \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}\}$, and $\gamma \in \{2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2^{0}\}$. Perform cross-validation for each pair of parameters Plot cross validation accuracy values as a surface plot (Hint: Matlab has `surf` function for this purpose). Give the best cross-validation accuracy and the corresponding C and γ values.

Re-train the model on the full training set using the best parameter combination, and test the resulting model on test set. Indicate the test-set accuracy you obtain and compare your result with the Linear SVM result. Output the decision values of test set as a file. Is Kernel SVM better, why?

Add the plots to your report, label the axes and add titles to plots, add a legend if necessary. Name your code as `hw2_q3a`(with `.m` if it is a Matlab code, `.py` if python ...etc.) for the linear kernel part, and `hw2_q3b` for RBF kernel part. As indicated in "How to Submit Programming Assignments" document, arguments should be given at runtime, for this part, the only argument is the file path to the dataset, and your code should do all the parts automatically.

4 Logistic Regression [30 pts]

In this part you will implement a Logistic Regression on the same dataset you used for previous question. Remember that in logistic regression, our goal is to learn a set of parameters by maximizing the conditional log likelihood of the data. Assuming you are given a dataset with n training examples and p features, the formula for the conditional log likelihood of the training data in terms of the the class labels $y^{(i)}$, the features $x_1^{(i)}, \ldots, x_p^{(i)}$, and the parameters $w_0, w_1, \ldots, w_p$, where the superscript (i) denotes the sample index is given below. This will be your objective function for gradient descent.

$$\begin{aligned} l(w_0, w_1, \ldots, w_p) &= log \prod_{i=1}^{n} P(y^{(i)}|x_1^{(i)}, \ldots, x_p^{(i)}; w_0, w_1, \ldots, w_p) \\ &= \sum_{i=1}^{n} \left[y^{(i)}(w_0 + \sum_{j=1}^{p} w_j x_j^{(i)}) - log(1 + exp(w_0 + \sum_{j=1}^{p} w_j x_j^{(i)})) \right] \end{aligned}$$

The partial derivative of the objective function with respect to w_0 and with respect to an arbitrary w_j, are given below, you will use these to update your parameter weights according to Gradient

Descent algorithm.

$$\frac{\sigma f}{\sigma w_0} = \sum_{i=1}^{n} \left[y^{(i)} - \frac{exp(w_0 + \sum_{j=1}^{p} w_j x_j^{(i)})}{1 + exp(w_0 + \sum_{j=1}^{p} w_j x_j^{(i)})} \right]$$

$$\frac{\sigma f}{\sigma w_j} = \sum_{i=1}^{n} x_j^{(i)} \left[y^{(i)} - \frac{exp(w_0 + \sum_{j=1}^{p} w_j x_j^{(i)})}{1 + exp(w_0 + \sum_{j=1}^{p} w_j x_j^{(i)})} \right]$$

using these functions first implement Gradient Descent for optimization with constant learning rate and then use it to implement your logistic regression classifier. Select the best learning rate by cross validation and plot the different train and test accuracy over number of iterations and give the best accuracy obtained over your test dataset.
In this part while you can use libraries for simple arithmetics, you can't use a library for implementing Logistic Regression classifier or Gradient Descend algorithm.

NOTES

- This homework will be graded by your TA, Iman Deznabi. Please ask the clarifications on Moodle, for other things you may ask questions at his office hours or by e-mail.
- Do not forget to add a **README** file that contains execution details for your code.

Arda Cortot BATI

1) a) T=0, No pruning will be made

There are 3 possible boundaries for x_1 feature: $x_1 = 1.5$, 4 and 6.5

" " 3 possible boundaries for x_2 feature: $x_2 = 1.5, 2.5$ and 4

To decide on which criteria to split, information gain after each possible boundary split should be calculated

First of all, $H_{parent} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$, this will always remain the same.

$\boxed{(H_{c_1} \to <, H_{c_2} \to > \text{ boundaries})}$

relative frequency

For $x_1 < 1.5$ or $x_1 > 1.5$ Boundary

$H_{c_1} = -\log_2 1 = 0$, $H_{c_2} = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.81$, $IG = H_{parent} - p_1 H_{c_1} - p_2 H_{c_2}$

(< Boundary) (> Boundary)

$= 1 - \frac{1}{4}\times 0 - \frac{3}{4}\times 0.81 = \underline{0.39}$

For $x_1 < 4$ or $x_1 > 4$ Boundary

$H_{c_1} = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.92$, $H_{c_2} = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.92$, $IG = 1 - \frac{1}{2}\times 0.92 - \frac{1}{2}\times 0.92$

$= 0.08$

For $x_1 < 6.5$ or $x_1 > 6$ Boundary

$H_{c_1} = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.81$, $H_{c_2} = -\log_2 1 = 0$, $IG = 1 - \frac{1}{4}\times 0 - \frac{3}{4}\times 0.81 = 0.39$

$x_2 \lessgtr 4$ Boundary

$H_{c_1} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, $H_{c_2} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, $IG = 1 - \frac{1}{2}\times 1 - \frac{1}{2}\times 1 = \underline{0}$

$x_2 \lessgtr 2.5$ Boundary, $H_{c_1} = \frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1 = H_{c_2}$, $IG = 1 - \frac{1}{3}\times 1 - \frac{2}{3}\times 1 = \underline{0}$

$x_2 \lessgtr 1.5$ Boundary, $H_{c_1} = 1$, $H_{c_2} = 1$, $IG = 0$

— The highest information gains are from $x_1 \lessgtr 1.5$ and $x_1 \gtrless 6.5$, which have the same gains. One can be chosen randomly. Let's choose $x_1 \gtrless 1.5$ as the first decision condition.

For $x_1 < 1.5$, the node contains all #'s. The second node is impure so it will be split next. The IG of the boundaries before will be calculated for this node (apart from $x_1 \lessgtr$ Boundary which is already used.)

next page

$-H_{node\,2} = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.81 \rightarrow$ Will remain constant.

For $x_1 \lessgtr 4$, $H_{c_1} = -\log_2 1 = 0$, $H_{c_2} = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.92$,

$IG = 0.81 - \frac{1}{4}\times 0 - \frac{3}{4}\times 0.92 = 0.12$

For $x_1 \lessgtr 6.5$ $H_{c_1} = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, $H_{c_2} = -\log_2 1 = 0$

$IG = 0.81 - \frac{1}{2}\times 1 - \frac{1}{2}\times 0 = 0.31$

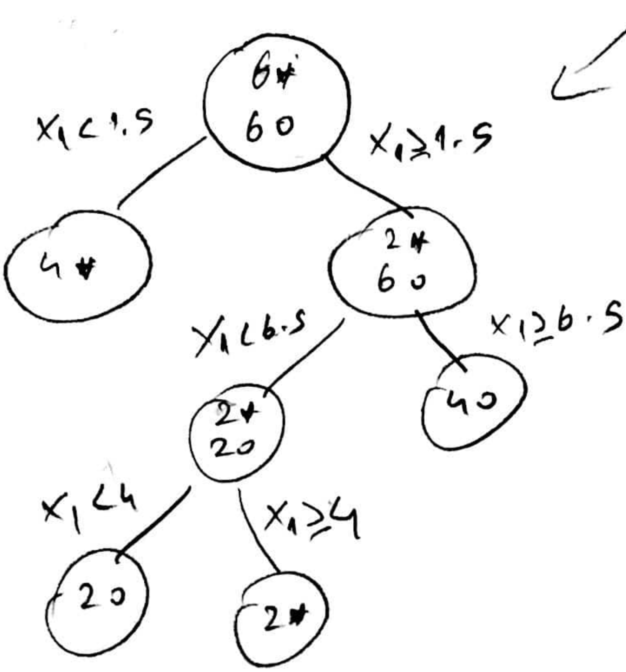
For $x_2 \gtrless 1.5$ $H_{c_1} = \log_2 1 = 0$, $H_{c_2} = -\ \log_2\frac{2}{7} - \frac{5}{7}\log_2\frac{5}{7} = 0.86$

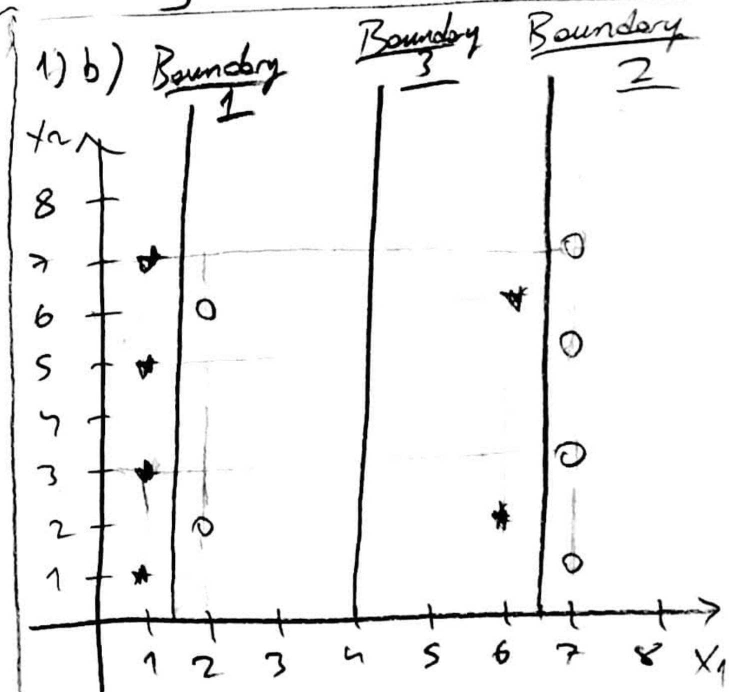
$IG = 0.81 - \frac{1}{8}\times 0 - \frac{7}{8}\times 0.86 = 0.0575$

For $x_2 \gtrless 4$, $H_{c_1} = H_{c_2} = -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} = 0.81$, $IG = 0$

For $x_2 \gtrless 1.5$

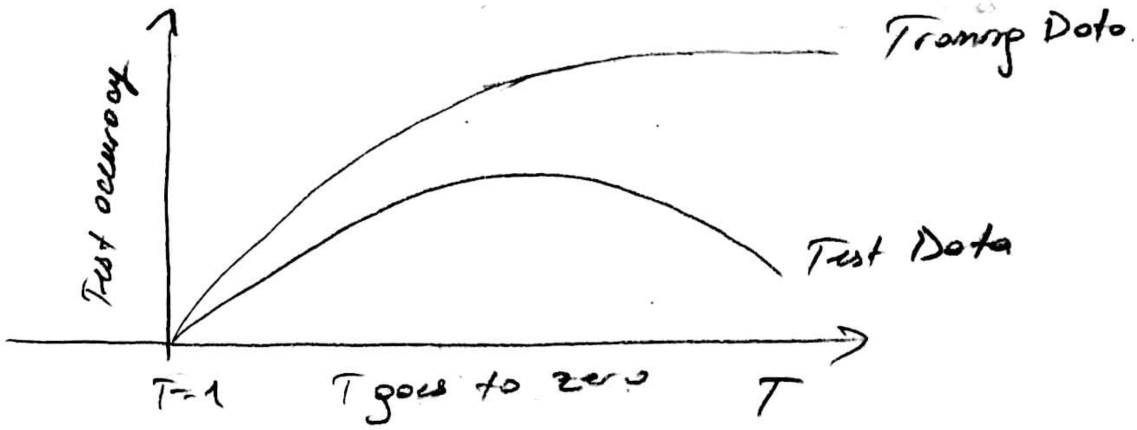
The boundary with the highest IG for Node 2 is $x_1 \gtrless 6.5$ Boundary, with this boundary, Node 2 splits into two child nodes, $x_1 > 6.5$ and $x_1 < 6.5$. For $x_1 > 6.5$ all elements are 0's, it is pure. The child $x_1 < 6.5$ contains 2 *'s and 2 0's. For this node, IG can be calculated again but $x_1 \gtrless 4$ is easily seen as a valid boundary which gives the maximum IG. (Refer to part (b) to see this more clearly.) The resulting tree is as follows:





next page

1) c) Very large data set, T changes from 1 —> 0 (on the X-axis)



— For T close to 1, the model will be less complex. There will be underfitting. The model will have lower accuracies for both train and test data.

— For T between a certain optimal threshold, both train and test data will have good accuracy, the model will be more generalized

— For T close to 0, there will be overfitting. Training data will give the highest test accuracy while Test data will start to give lower test accuracies than before. The model will become to specialized.

Question 2

a) Decision boundary is drawn on the figure given in the question.

b) $\bar{w}$ should be perpendicular to the decision boundary hyperplane.

Also $\gamma = \frac{1}{\|\bar{w}\|}$ for margin size.

— The decision boundary line is $y = x-2$, direction vector $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$

$\bar{w}$'s direction should be $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$, so that $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}) \cdot (\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}) = 0$

— From the lines drawn on the figure, margin can be seen to be $\gamma = \sqrt{2}$.

$\gamma = \sqrt{2} = \frac{1}{\|\bar{w}\|}$, $\|\bar{w}\| = \frac{1}{\sqrt{2}}$, then $\bar{w} = (\frac{1}{2}, -\frac{1}{2})$ which satisfies $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$ direction vector mentioned before.

next page

Question 2) b) continued

$-1-[2]$

to find b, we can put our support vectors in the $\bar{w}\bar{x_i}+b=y_i$ formula, where $\bar{x_i}$ is a support vector and y_i is its label.

We have (4,4) and (6,2) support vectors, assumed to be labeled +1 and -1 respectively. $y_i \in \{+1,-1\}$

$$b=-1-\begin{bmatrix}\frac{1}{2} & -\frac{1}{2}\end{bmatrix}\begin{bmatrix}4\\4\end{bmatrix}=-1, \quad y_1=-1, \quad (4,4)$$

$$b=1-\begin{bmatrix}\frac{1}{2} & -\frac{1}{2}\end{bmatrix}\begin{bmatrix}6\\2\end{bmatrix}=-1, \quad y_2=1, \quad (6,2)$$

Q2) c) The weight of each point can be found from Lagrangian α_i.

We know $\bar{w}=\sum_{i=1}^{2}\alpha_i y_i x_i$ and $\bar{w}=\sum_{i=1}^{2}\alpha_i y_i=0$ from the conditions set for the primal Lagrangian.

$$\bar{w}=\sum_{i=1}^{2}\alpha_i y_i x_i \Rightarrow \left(\frac{1}{2},-\frac{1}{2}\right)=\alpha_1.(-1).(4,4)+\alpha_2.+1.(6,2)$$

$$\frac{1}{2}=-4\alpha_1+6\alpha_2, \quad -\frac{1}{2}=-4\alpha_1+2\alpha_2 \Rightarrow \alpha_1=\frac{1}{4}, \alpha_2=\frac{1}{4}$$

d) The weight of the point will increase, because the margin is lowered, it will increases, and the point is still a support vector. (The point becomes even more critical in defining the decision boundary.)

$-1+\frac{2}{4}$

Question 3 Part b)

The accuracy of Linear SVM on test data : 0.9873
The " " Gaussian " " " " : 0.9991

The Gaussian Kernel SVM has slightly higher accuracy on the test data. This may be because it uses a Gaussian Kernel which generalizes to real-life statistics better than linear functions. However the accuracy difference is very small to make a solid argument.