1. **Short Questions**. Answer each questions with at most two sentences, and/or a picture. For the (true/false) questions, if the answer is true, provide a short justification. If false, explain why or give a small counter example. No point if no proper explanation is provided.

a. (2pts) (true/false) We can use linear regression to learn the parameters for the model $y = w_1 x_1 x_2 + w_2 x_2 \log x_1$.

True. Treat $(x_1 x_2, x_2 \log x_1)$ as the new features. then it is a simple linear model.

b. (2 pts) (true/false) We can use linear regression to learn the parameters for the model $y = x_1^{w_1} + x_2^{w_2}$.

False. No transformation can make this linear. Note taking log on both sides will be $\log y = \log(x_1^{w_1} + x_2^{w_2})$

d. (3 pts) Why does the kernel trick allow us to solve SVMs with high dimensional feature spaces, without signicantly increasing the running time?

A Kernel function is equivalent to dot product in a high dimensional transformed space without explicitly computing the transformation.

d. (4 pts) Consider the following data set:

o +

\+ o

Circle all the classifiers that will achieve zero training error on this data set. (maybe more than one).

i. Logistic Regression
ii. Depth-2 Decision tree
iii. Support vector machine with linear kernel
iv. Support vector machine with quadratic kernel

e. (4pts) Assume we computed the parameters for a Naive Bayes classifier. How can we use these parameters to compute $P(x)$ for a given input vector $x = [x_1, x_2, ..., x_d]^T$?

$$P(x) = \sum_y P(x, y) = \sum_y P(x|y)P(y) = \sum_y \prod_{i=1}^{d} P(x_i|y)P(y).$$

2. **Linear Regression and regularization**. We have a regression problem with the target y and a single input feature x. We know that y is a polynomial function of x but do not know the exact order of the polynomial except that it is ≤ 4. Answer the following questions.

a. (4pts) Given a training set, consider the following strategy for learning the polynomial function. For each $k \in \{1, 2, 3, 4\}$, we learn a linear regression model $\hat{y} = w_0 + w_1 x + \cdots + w_k x^k$ by minimizing the Sum of Squared Error (SSE) $\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$ on the training data. We then pick the model that achieves the lowest training SSE. Will this strategy find the correct order k? Briefly justify you answer.

No. As we increase K, SSE will monotonically decrease.

b. (4pts) Now we will focus on the 4-th order model $\hat{y} = w_0 + w_1 x + \cdots + w_4 x^4$ and learn the parameters by minimizing the following objective with L_1 regularization:

$$\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \lambda \sum_{j=1}^{4} |w_j|$$

As the value for λ increases, what will happen to the training SSE of the learned model and why?

SSE will increase because with large λ we will focus on making $\sum_j |w_j|$ small, and reducing complexity of the model, leading to larger SSE.

c. (4pts) Continue from [b], the true model is $y = x + x^2$. As we gradually increase λ to a very large value, which of the following do you expect to happen to the learned parameters $w_1, ..., w_4$? (may have multiple correct ones.) Briefly justify your answer.

i. The changes to the weights are unpredictable.
ii. All weights will become smaller at the same time.
iii. w_3 and w_4 will decrease to zero first. ✓
iv. Eventually all weights will decrease to zero as λ gets to be large enough. ✓

W_3 and W_4 does not contribute to minimizing SSE, so making them zero will improve the objective as we make λ larger. When λ is large enough, $\sum_j |w_j|$ will overwhelm the SSE part, and the optimal solution will be making all weights zero.

3. Support vector machines. For the data set shown below, we apply soft margin SVM.

a (2.5pts) Where would the decision boundary be for a very small c value, e.g., $c = 0.00001$?

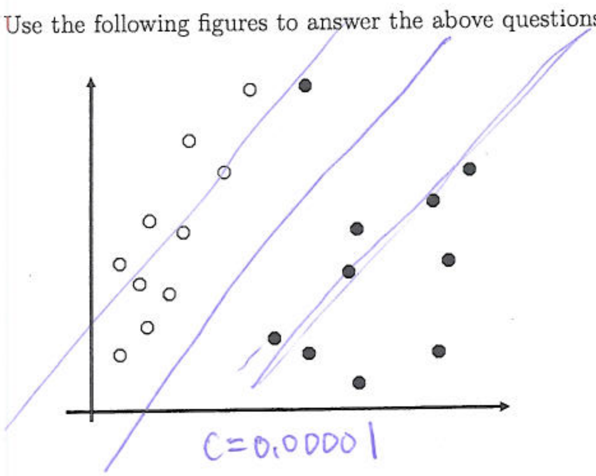
b (2.5pts) Where would the decision boundary be for a very large c value, e.g., $c = 100000$?

Now we will add another **positive** example (solid dot) to the training set.

c (2.5pts) Where could you place it so that it will not influence the learned decision boundary for a very large c value.

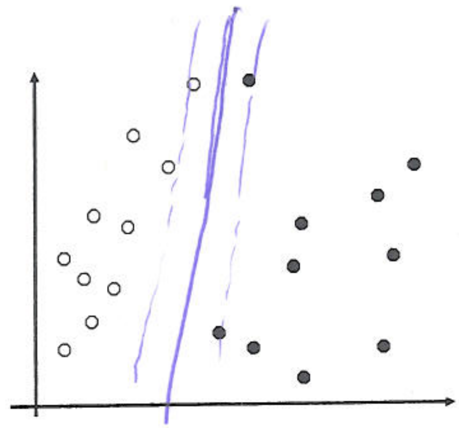
d (2.5pts) Where could you place it so that it will significantly influence the learning decision boundary for a very large c value.

Use the following figures to answer the above questions.



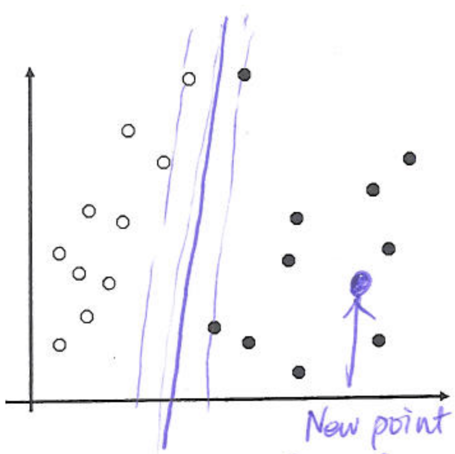
C=0.00001

(a)



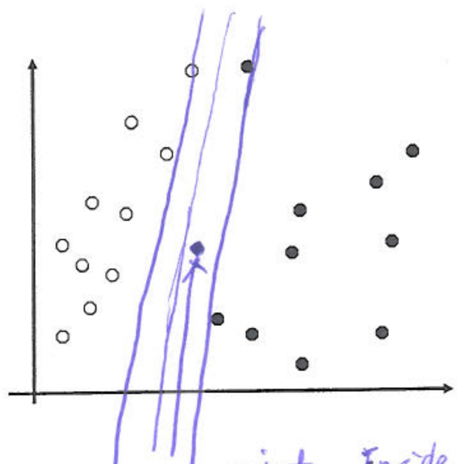
C= 100000

(b)



New point Away from decision boundary. No impact on it.

(c)



New point. Inside the tube, forcing the decision boundary to change because it will incur some additional loss.

(d)

4. Use the following data to build a decision tree to predict whether a student will be lazy (L) or diligent (D) based on weight (Normal or Underweight), eye color (Amber or Violet) and the number of eyes(2, 3 or 4).

Weight	Eye Color	Num. Eyes	Output
N	A	2	L
N	V	2	L
N	V	2	L
U	V	3	L
U	V	3	L
U	A	4	D
N	A	4	D
N	V	4	D
U	A	3	D
U	A	3	D

You may find these useful:

$\log_2 0.2 = -2.32 \quad \log_2 0.4 = -1.32$

$\log_2 0.6 = -0.73 \quad \log_2 0.8 = -0.32.$

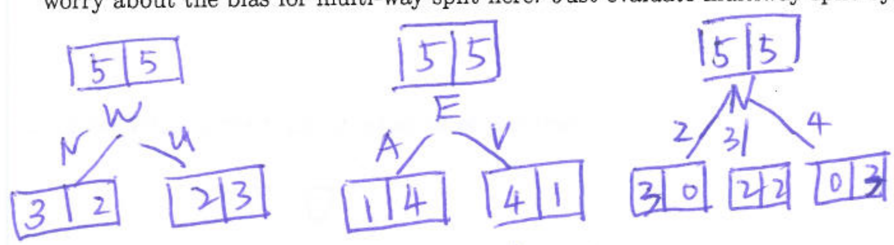
a. (3pts) What is the conditional entropy $H(Eyecolor|Weight = N)$?

Weight = N, Eyecolor = A .. 2 instances.

Weight = N, Eyecolor = V. 3 "

$$-\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.4\times1.32 + 0.6\times0.73 = 0.966$$

b. (4pts) What attribute will information gain select to place at the root of the tree? (Note: we do not worry about the bias for multi-way split here. Just evaluate multiway split by its information gain.)

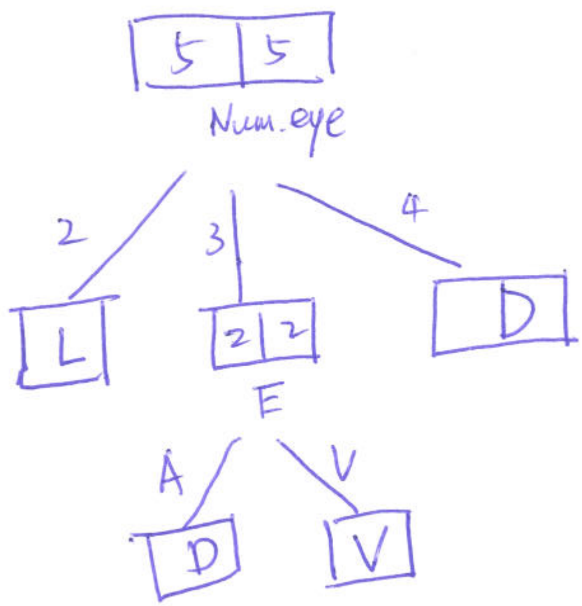


$$H(L|E) = \frac{1}{2}\cdot\left(-\frac{1}{5}\log\frac{1}{5} - \frac{4}{5}\log\frac{4}{5}\right)\times 2 = 0.2\times2.32 + 0.8\times0.32$$
$$= 0.464 + 0.256 = 0.72$$

$$H(L|N) = \frac{6}{10}\cdot 0 + \frac{4}{10}\cdot 1 = 0.4 \quad ✓$$

Choose Num. eye.

c. (4pts) Draw the full decision tree learned from this data (no pruning or early stopping).



d. (2pts) What is the training set error of this tree?

0.