CS 6375 Machine Learning Endterm Examination

University of Texas at Dallas

12/02/2020

Finals Solutions

Name: _____

NetID: _____

Question	Topic	Points
1	Short Answers	20
2	True/False Questions	8
3	Linear Regression	15
4	Clustering	15
5	Decision Trees and Ensembles	12 + 6 Bonus
6	Neural Networks	8
7	Cost Functions	12
8	Learning Theory	10
Total		100 + 6 Bonus

Instructions:

1. This examination contains 18 pages, including this page.
2. You have **two and a half (2.5) hours** to complete the examination.
3. Either you can use this paper or separate set of sheets to fill in your answers. Write clearly so we can understand your handwriting. Please scan the answers once you are done (you can also take a photograph once you are done) and upload the scanned copy to eLearning. You can also use a tablet device (like an Ipad) to write if you prefer.
4. Please do not search online for answers to the questions. If the answers are similar to something available online, you will get zero points on this examination.
5. The end-term examination has to be done individually by everyone. If someone copies, the entire group of students involved will get a zero.
6. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult one.
7. All the Best!!

Question 1: Short Answers

[20 pts] Please provide short and clear answers for the questions below.

(a) (2 points) For linearly separable data, can a small slack penalty ("C") hurt the training accuracy when using a linear SVM (with no kernel)? If so, explain how and if not, why not?

Yes. training error increases when loss fn is optimized with regularization more reg ⇒ smaller "w" which could favor generalization over train error.

(b) (4 points) Consider running AdaBoost with Multinomial Naive Bayes as the weak learner for two classes and k binary features. After t iterations, of AdaBoost, how many parameters do you need to remember? In other words, how many numbers do you need to keep around to predict the label of a new example? Assume that the weak-learner training error is non-zero at iteration t. Don't forget to mention where the parameters come from.

- At each iteration of Adaboost, we need to remember $h_t(x)$, α_t : $y = \text{sign}\left(\sum_t \alpha_t h_t(x)\right)$
- if Multinomial NB is the weak classifier, $2k$ parameters for $P(x_i|y=0)$ & $P(x_i|y=1)$ $\forall i$ & 1 parameter for $P(y=1)$

Total : $(2k+2)t$ parameters

(c) (4 points) Would you stop boosting if the following happens? Justify your answer with at most two sentences each question. Each part is 2 points.

- Part 1: The error rate of the combined classifier on the training set is 0 No
- Part 2: The error rate of the current weak classifier on the weighted training data is 0 Yes

Part 1: No since boosting is robust to overfitting even if train error = 0, it might continue reducing test error till algo stops

Part II - true since $\alpha_t = +\infty$ and weights will be zero. Existing boosted model is good enough & we will stop.

(d) (2 points) Imagine we are running stochastic gradient descent, and we determine the stopping criteria based on the validation set. Suppose the stopping criteria is if the validation set error increases (i.e. stop SGD as soon as the validation set error starts increasing). Is this a good stopping criteria? If not, how would you fix it?

No. since val error might increase while training.

Better strategy is have a "patience parameter" - wait for 'k' epochs to make sure val-error does not keep increasing (k = hyperparam)

(e) (3 points) Given n linearly independent feature vectors in n dimensions. Then, for any assignment to the binary labels, is it possible to always construct a linear classifier with weight vector w which separates the points. Assume that the classifier has the form $sign(wx)$. Also, note that a square matrix

if $Xw = y \Rightarrow sign(Xw) = y$

Since X is invertible we can solve for w. Also can be done by considering LR. so yes, we can construct such a linear classifier!

composed of linearly independent rows is invertible.

(f) (3 points) Construct a one dimensional classification dataset for which the Leave-one-out cross validation error of the One Nearest Neighbors algorithm is always 1 (i.e. 100% error). Stated another way, the One Nearest Neighbor algorithm never correctly predicts the held out point.

for any n, create alt-configs
of $\{t_1, -, t_2, -, \ldots\}$
LOOCV 1NN will get 0 accuracy!

(g) (2 points) Imagine we have a binary classification problem where the positives are 10% and negatives are 90%. What will be the accuracy of a classifier which only predicts 0? What will be the AUC of such a classifier?

Accuracy = 90%
AUC = 0 (since random - will always be ranked above random +)

Question 2: True or False with Explanations

[8 pts] Each question below is for 1 point. Please do not just write true or false. You also need to provide explanations. Just true or false will yield no points.

(a) We can tune the regularization parameter λ for regularized logistic-regression on the test set and that will give us a good indication of the generalization performance.

No. Tune on val set & not test-set. This is cheating!!

(b) In SVMs, the values of α_i for non-support vectors are 0

True.

(c) In the case of binary classification, the AUC of a random classifier is 0

False, AUC (Random) = 0.5

(d) Cross validation will guarantee that our model does not overfit.

False.
Need held out val set

(e) Given a binary classification scenario with Gaussian class conditionals and equal prior probabilities, the optimal decision boundary will be linear.

False. Quadratic / Elliptical.

(f) In the primal version of SVM, we are minimizing the Lagrangian with respect to w and in the dual version, we are minimizing the Lagrangian with respect to α.

False (max w.r.t α)

(g) The VC dimension of a classifier always equals the number of parameters of the classifier.

False

(h) Since classification is a special case of regression, logistic regression is a special case of linear regression.

False.

Question 3: Linear Regression

[15 pts] **Background:** In this problem we are working on linear regression with regularization on points

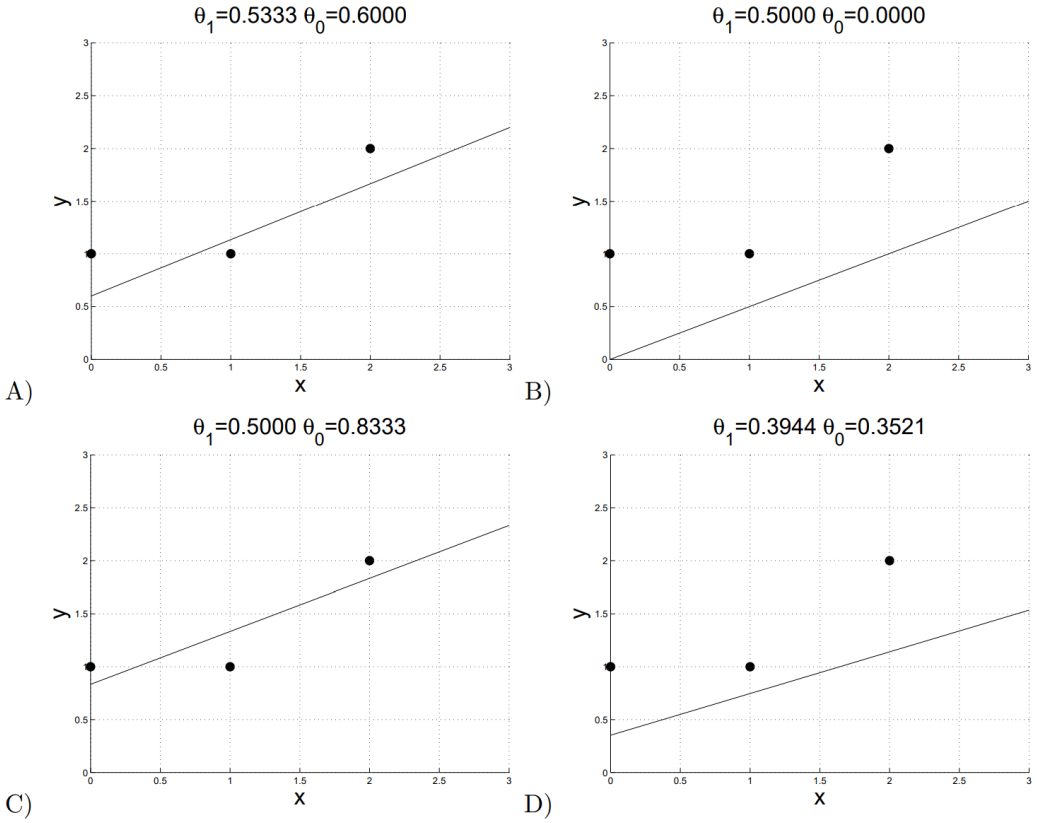


Figure 1: Linear Regression results for different regularizations

in a 2-D space. Figure 1 plots linear regression results on the basis of three data points, (0,1), (1,1) and (2,2), with different regularization penalties. Recall, that linear regression involves the following optimization problem in 2D:

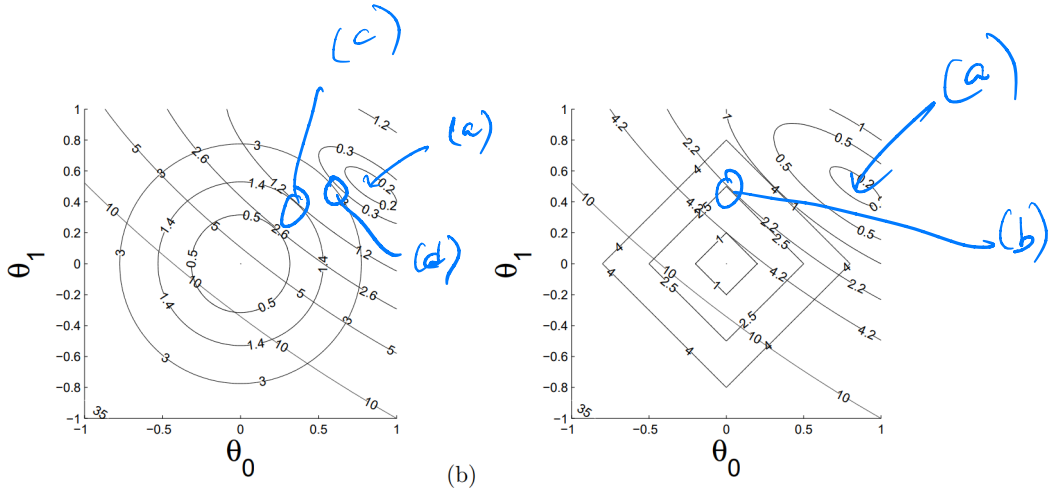
$$\min_{\theta_0,\theta_1} \sum_{i=1}^{n} (y_i - \theta_1 x_i - \theta_0) + R(\theta_0, \theta_1) \tag{1}$$

where R could either be L1 or L2 Regularization. However, instead of computing the derivatives to get a minimum value, we could adopt a geometric method. In this way, rather than letting the square error term and the regularization penalty term vary simultaneously as a function of θ_0 and θ_1, we can fix one and only let the other vary at a time. Having a upper-bound, r, on the penalty, we can replace $R(\theta_0, \theta_1)$ by r, and solve a minimization problem on the square error term for any non-negative value of r. Finally, we get the minimum value by enumerating over all possible value of r. That is equation 1 is equivalent to:

$$\min_{r \geq 0} \{ \min_{\theta_0,\theta_1} [(y_i - \theta_1 x_i - \theta_0) \mid R(\theta_0, \theta_1) \leq r] + r \} \tag{2}$$

In Figure 2, we plot the square error term by ellipse countours. The circle contours in Fig 2(a) plots a L-2 penalty with $\lambda = 5$, whereas the square contours in Fig 2(b) plots a L-1 penalty with $\lambda = 5$. To further explain how it works, the solution to:

$$\min_{\theta_0,\theta_1} [(y_i - \theta_1 x_i - \theta_0) \mid R(\theta_0, \theta_1) \leq r] \tag{3}$$



(a) (b)

Figure 2: Contour plots of the decomposition for the linear regression problem with (a) L-2 regularization or (b) L-1 regularization where the ellipsis correspond to the square error term, and circles/squares correspond to the regularization penalty term.

is the height of the smallest ellipse contour that is tangent with (or contained in) the contour that depict $R(\theta_0, \theta_1) = r$. The desired (θ_0, θ_1) are the coordinates of the tangent point.

Questions

(a) • Please assign each plot in Figure 1 to one (and only one) of the following regularization methods. You can get some help from Figure 2. Please answer A, B, C or D. Please explain your answer.

(a) (3 points) No regularization

$\theta_0 = 0.83$, $\theta_1 = 0.5$

C

See Figure 2(a) or 2(b)

(b) (3 points) L1 regularization with $\lambda = 5$

(Note Fig 2 shows B with $\lambda = 5$)

B

Figure 2b

Options: $4+1$, $2.2+2.5$ or $1+4.2$

5 5.2 4.2

$\hat{\theta} \simeq (0.5, 0)$

(c) (3 points) L2 regularization with $\lambda = 5$

D $\theta_0 \approx 0.79$, $\theta_1 \approx 0.75$

options: $\underbrace{1.4 + 1.2}_{2.8}$, $\underbrace{2.6 + 0.5}_{3.1}$, $\underbrace{3 + 0.3}_{3.3}$

(d) (3 points) L2 regularization with $\lambda = 1$

options: $\underbrace{1.2 + \frac{1.4}{5}}_{1.48}$, $\underbrace{0.3 + \frac{3}{5}}_{0.9}$, $\underbrace{2.6 + \frac{0.5}{5}}_{2.7}$

A

- (3 points) If we have much more features and we want to perform feature selection while solving the LR problem, which kind of regularization method do we want to use? What about λ?

L1 regularization with high λ

Tune λ till you get the desired features

Question 4: Clustering

[15 pts] This question has two parts. The first part is for 9 points and second is for 6 points.

- Part 1 (9 Points): Consider the following clustering method called Threshold Clustering. It receives two parameters: an integer k and a real number t. Similar to k-means, it starts by selecting k representatives and assigns each training instances to the cluster of the closest representatives. During the assignment step, however, if the distance of a training instance to its closest representatives is greater than the input threshold t, then this training instance becomes a new representative. During the same assignment step, remaining points can be assigned to these new representatives. After all the training instances have been assigned to a cluster, new representatives are calculated by averaging each cluster. The process is then repeated until the cluster assignments do not change.

  - (5 points) When does the threshold clustering produce more, the same number, or fewer clusters than k-means, assuming that the k initial centers are the same for both? When will the clusterings produced by threshold clustering and k-means be identical? Explain.

    Small threshold: More representatives are likely to be formed. ∴ threshold clustering will obtain more centroids
    Large threshold, likely to be similar to k-means

  - (4 points) Which of the two methods (regular k-means or threshold Clustering) will be best at dealing with ourliers? I.e. data instances that are very far away or different to other instances in the dataset. Explain.

    Threshold Clustering will be better in dealing with outliers since it will create separate clusters for them.
    k-means will tend to combine outliers into normal point if k is small!

- Part 2 (6 points): The are six different datasets shown in Figure 3, and only one of them is obtained using k-means (and running it till convergence). In each case, identify which one is from k-means and why.

  Idea: k-means will produce output s.t each point must be closer to its own representative.

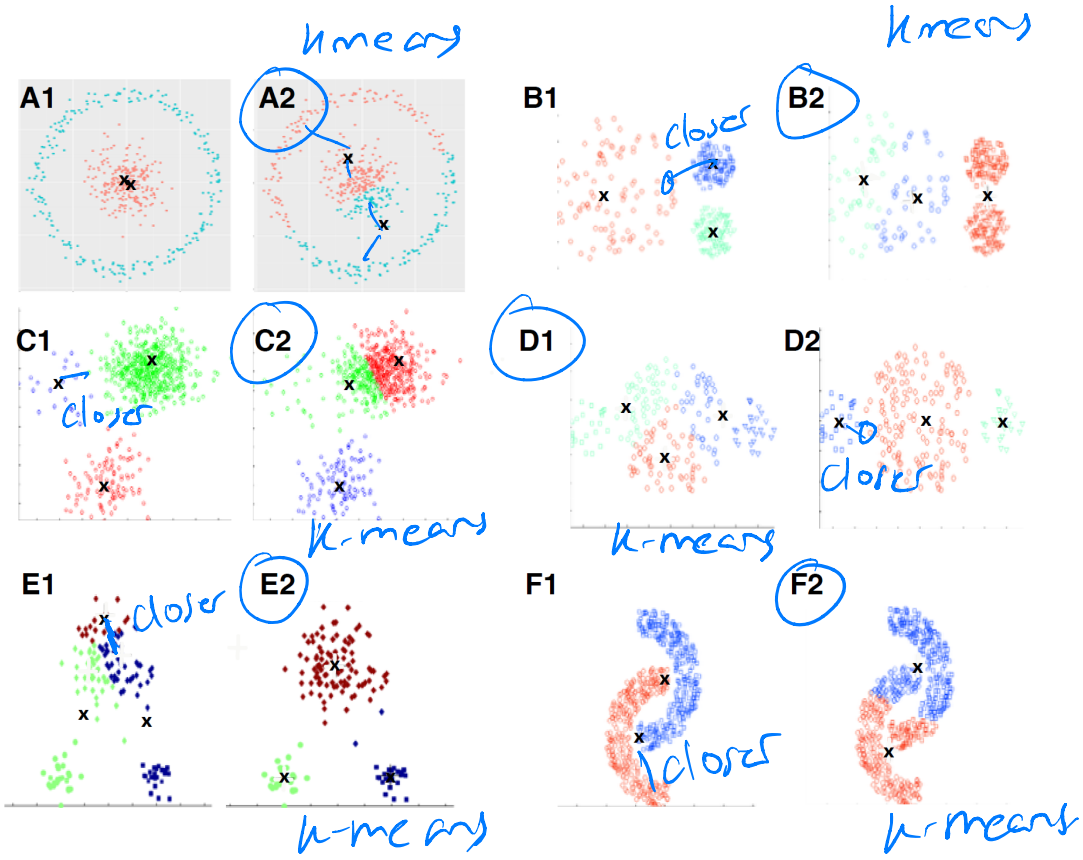


Figure 3: Clusterings for Datasets A, B, C, D, E and F.

Question 5: Decision Trees and Ensembles

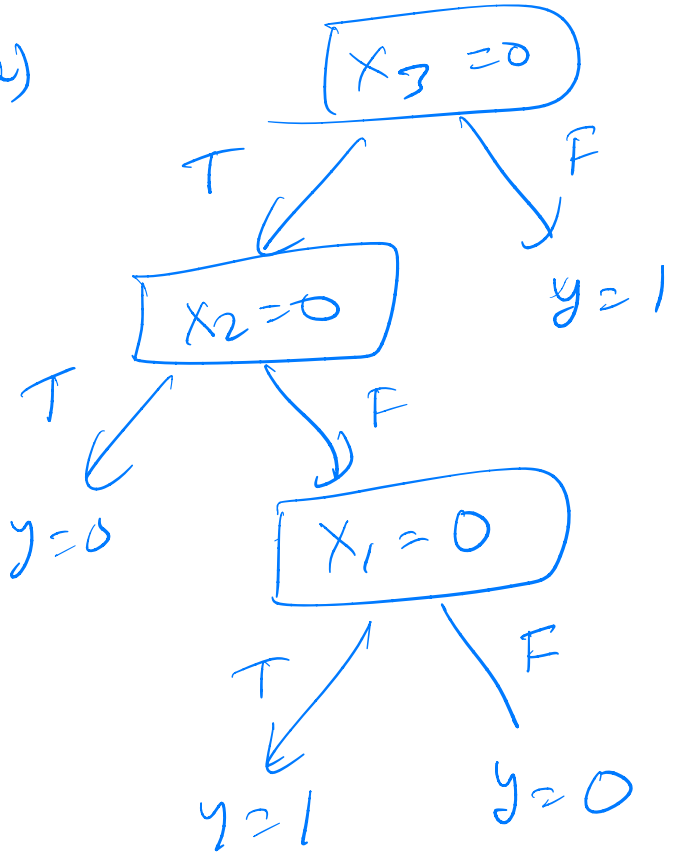
[12 + 6 bonus pts] This question will cover decision trees, random forests and gradient boosted trees. Consider a dataset with three features x_1, x_2, x_3. Consider the following dataset consisting of feature and label pairs: $[\{(0,0,1),1\},\{(0,1,0),1\},\{(1,1,1),1\},\{(1,1,1),1\},\{(0,0,0),0\},\{(0,0,0),0\},\{(1,1,0),0\}]$. Note that there are 7 data-points here. Feel free to use a calculator to calculate the exact values (entropies and information gain). **You only need to answer parts (a) and one of parts (b or c). In case you answer all three parts correctly, you can receive 6 bonus points.**

(a) (6 points) In this part, we will compute a single decision tree. Which variable will you choose as the root node? Draw the resulting decision tree and show your workings.

(b) (6 points) Suppose we were to use a random forest of 3 decision trees, with each tree being of depth 2. Provide one example random forest which would come out of this. Show workings

(c) (6 points) Suppose we were to use gradient boosting to obtain an ensemble of 3 decision trees. Provide the algorithm as well as the final ensemble you would obtain. Assume the depth of each tree is 2, and use the square loss for boosting. Show workings.

(a)



(b) Pick any subset of random features to form 3 Decision trees

e.g (X_1, X_2), (X_2, X_3), (X_1, X_2)

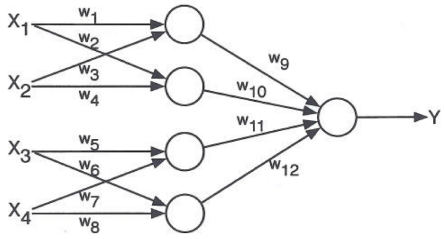
You can also take subset of datapoints

It is ok to provide one specific Random Forest

(c) Similarly for part (c), do boosting start with Depth 2 DT for part (a) & Construct y as residual & repeat.

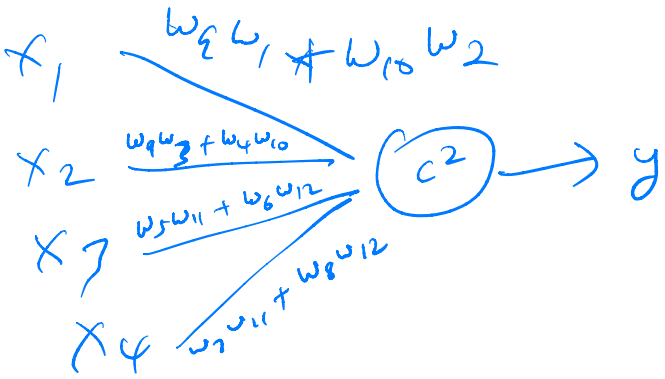
Question 6: Neural Networks

[8 pts] Assume that we have the following neural network with linear activation units. The output of each unit is a constant c multiplied with the weighted sum of inputs.



(a) (4 points) Is it possible for any function that can be represented by the above network to be represented using a single unit network? If so draw such a network including the weights and the activation function. If not, briefly explain why not.

yes



(b) (4 points) Is it possible for any function that can be represented by the above network to be represented using a linear model? If so, provide the equation for Y. If not, explain why not.

$$y = c^2(w_9w_1 + w_{10}w_2)x_1 + c^2(w_9w_3 + w_{10}w_4)x_2 + c^2(w_5w_{11} + w_6w_{12})x_3 + c^2(w_7w_{11} + w_8w_{12})x_4$$

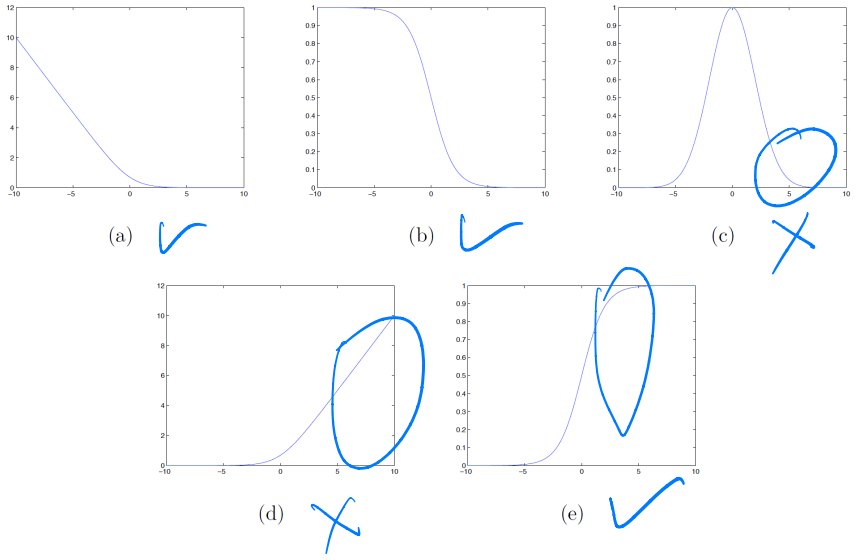
Question 7: Loss Functions

[12 pts] Generally speaking, a classifier can be written as $H(x) = sign(F(x))$, where $H(x) : R^d \to \{-1, +1\}$, and $F(x) : R^d \to R$. To learn F, a common strategy is to minimize a loss function on the training set:

$$\min_F \sum_{i=1}^{n} L(y^i F(x^i)) \tag{4}$$

Here L is a function of $yF(x)$. For linear classification, $F(x) = w_0 + \sum_{j=1}^{d} w_j x_j$

(a) (4 points) Which loss functions below are appropriate to use in classication and why? For the ones that are not appropriate, explain why not. In general, what conditions does L have to satisfy in order to be an appropriate loss function? The x axis is $yF(x)$, and the y axis is $L(yF(x))$.



(a) ✓ (b) ✓ (c) ✗ (d) ✗ (e) ✓

Property : Loss is low if $yF(x) \geq 0$

Loss High if $yF(x) < 0$

& depends on how neg $yF(x)$ is

(b) (4 points) Of the above loss functions appropriate to use in classication, which one is the most robust

to outliers? Justify your answer.

(b) Flattens out when $yF(x) \ll -5$.

unlike (a)

So (b) is less affected by outliers

Compared to (a)

(c) (4 points) Let $F(x) = w_0 + \sum_{j=1}^{d} w_j x_j$. Give the general update rules for gradient descent to update the weights w (it is ok to provide the expression in terms of the derivative of L.

Use chain rule.

e.g. $$\frac{\partial L(yF(x))}{\partial w_i} = \frac{\partial L(y w_0 + y \sum_{i=1}^{d} w_i x_i)}{\partial w_i}$$

$$= \frac{\partial L(z)}{\partial z} \cdot \frac{\partial (y w_0 + y \sum_{i=1}^{d} w_i x_i)}{\partial w}$$

$$z = y w_0 + y \sum_{i=1}^{d} w_i x_i$$

$$\therefore \quad \frac{\partial L}{\partial w_0} = \frac{\partial L(z)}{\partial z} \cdot y$$

$$\frac{\partial L}{\partial w_i} = \frac{\partial L(z)}{\partial z} \cdot y x_i$$

} Perform GD updates on this

Question 8: Learning Theory

[10 pts] Answer each of the subquestions below.

(a) (2 points) Can the set of all rectangles in 2D (which includes non axis-aligned rectangles) shatter a set of 5 points. Explain

yes, e.g. pentagon

VC Dim of non-axis aligned rect is 7!

(b) (2 points) What is the VC dimension of k Nearest Neighbour classifier when $k = 1$. Explain.

0D Arbitrary large sets of points can be shattered

(c) (2 points) Which of the following is true if the VC dimension of a hypothesis class is $D < \infty$. a) There exists some set of D points shattered by the hypothesis class, b) All sets of D points are shattered by the hypothesis class, c) There exists a set of $D + 1$ points not shattered by the hypothesis, and d) No set of $D + 1$ points is shattered by the hypothesis class.

a & d

(d) (4 points) Consider the following formulas that bound the number of training examples necessary for successful learning:

$$m \geq \frac{1}{\epsilon}(\log(1/\delta) + \log|H|)$$

$$m \geq \frac{1}{2\epsilon^2}(\log(1/\delta) + \log(|H|))$$

$$m \geq \frac{1}{\epsilon}(4\log(2/\delta) + 8VC(H)\log(12/\epsilon))$$

For each of the below questions, pick the formula you would use to estimate the number of examples you would need to learn the concept. You do not need to do any computation or plug in any numbers. Explain your answer.

(a) Consider instances with two Boolean variables $\{X1, X2\}$, and responses Y are given by the XOR function. We try to learn the function $f : X \to Y$ using a 2-layer neural network.

Eqn 3 since hypothesis space |H| is ∞

(b) Consider instances with two Boolean variables $\{X1, X2\}$, and responses Y are given by the XOR function. We try to learn the function $f : X \to Y$ using a depth-two decision tree. This tree has four leaves, all distance two from the top.

Eq 1 since |H| is finite