CS760 Midterm Practice Questions

Machine Learning Overview and Evaluation: True/False

1. Unsupervised learning methods deal with instances without labels, and can reveal patterns of the data. $\boxed{True}$
2. In cross validation, we train the classifier using all of the data, and predict the classification of the left-out set. $\boxed{False}$ In cross validation, we train the classifier using all but one fold of the data.
3. High capacity models are more likely to overfit $\boxed{True}$

Machine Learning Overview and Evaluation: Short Answer

1. What is the hypothesis space bias and preference bias for the k-NN learner? Hypothesis space: Decomposition of space determined by nearest neighbors Preference Bias: Instances in neighborhood belong to same class (Supervised Learning 1)
2. Use the data from Table 1. What is the false negative and true negative rate? False Negative Rate (FNR): FN / (FN + TP) = 32 / (32 + 12) = 0.7272 Specificity (True Negative Rate): TN / (TN + FP) = 1 / (1 + 54) = 0.0181

	True Label	False Label
True Prediction	12	54
False Prediction	32	1

Table 1:

Neural Networks

1. Derive the derivative of ReLU activation function $ReLU(x) = max(0, x)$.

$$\frac{\partial}{\partial x}\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

And, undefined at x=0 since left derivative and right derivative are different.

2. Derive the derivative of hyperbolic tangent activation function $tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$.

$$\begin{aligned} \frac{d}{dx}\tanh(x) &= \frac{d}{dx}\left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right) \\ &= \frac{(e^x + e^{-x})(e^x + e^{-x}) - (e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2} \\ &= 1 - \frac{(e^x - e^{-x})(e^x - e^{-x})}{(e^x + e^{-x})^2} \\ &= 1 - \tanh^2(x) \end{aligned}$$

3. Compare sigmoid, hyperbolic tangent, and ReLU activation functions.
   - $\tanh(x) = 2\sigma(2x) - 1$
   - ReLU has a non-differentiable point, but sigmoid and hyperbolic tangent are differentiable at all points.
   - Sigmoid and hyperbolic tangent suffer from the vanishing gradient problem, but ReLU does not.
   - They have different ranges.

4. A fully connected feedforward neural network has input R2, a first hidden ReLU layer with 4 units, a second hidden ReLU layer with 3 units, and a single sigmoid output unit. The number of parameters in the neural network (including offset weights) = 30.

   Consider a fully connected feedforward neural network with input $\mathbb{R}^d$, a first hidden ReLU layer with h_1 units, a second hidden ReLU layer with h_2 units, and a single sigmoid output unit. Each hidden and output unit has an offset parameter. Input to the first hidden layer: $(1 + d)h_1$. First to the second hidden layer: $(1 + h_1)h_2$. To the output: $1 + h_2$. The total is the sum of these, i.e.,

$$(1 + d)h_1 + (1 + h_1)h_2 + 1 + h_2 = 3 \cdot 4 + 5 \cdot 3 + 4 = 31$$

Distances and Normalization

1. In a kNN, what are the situations in which you would use hamming distance and euclidean distance?

Given this dataset:

index	1	2	3	4	5	6	7
value	23	54	-12	912	6	7	12

Table 2:

1. What is the mean of the dataset? 143.14
2. What is the standard deviation? 339.6
3. What would be the dataset after standardization?

index	1	2	3	4	5	6	7
value	-0.353	-0.2624	–0.456	2.264	-0.403	-0.40088	-0.3861

Table 3:

Decision Trees

1. What is the drawback of using InfoGain when dealing with features that have many outcomes, and how does GainRatio address this issue? Information Gain favors features with many outcomes, which can lead to biased feature selection. Gain Ratio normalizes the Information Gain by considering the potential randomness in the feature, making it a fairer measure when dealing with features of varying cardinalities.
2. Why is it important to maximize mutual information in the context of choosing a split for decision tree construction, and what does it help achieve in terms of reducing uncertainty? Maximizing mutual information when choosing a split for a decision tree helps find the feature that gives the most helpful information for making accurate predictions. This reduces uncertainty in the data by separating it into more distinct groups, making it easier for the tree to make decisions.

Data Augmentation

1. How can data augmentation techniques like substitution and back-translation be applied to text data to enhance natural language processing tasks? Substitution and back-translation are tricks to make NLP models better. Substitution swaps words, and back-translation involves translating text back and forth. These tricks add variety to the training data, making

models understand language better and work well for tasks like translation and sentiment analysis.

2. What are similar techniques for images? Rotation/translation. Color augmentation, cropping/resizing, adding noise.
3. List the methods used to prevent overfitting. Regularization prevents overfitting. Examples are L1 and L2 regularization

Classification and Regression

1. What is the difference between classification and regression Classification: Output is categorical or discrete. Objective is to assign input to a category. Example uses are spam detection or image recognition. Regression: Output is numerical. Objective is to predict a value. Example uses are price prediction.
2. For linear regression, explain the concept of the cost function (mean squared error) and how it is minimized to find the best-fitting line. Cost function quantifies the error or the difference between the predicted values of the linear regression model and the actual observed values in the training data. The goal of linear regression is to find the best-fitting line that minimizes this cost function.
3. When using k-nearest neighbors (KNN) for classification, how does the choice of k (the number of neighbors) impact the decision boundary. Smaller k decision boundary is more sensitive to noise and individual data points values. Larger k has a smoother and more generalized decision boundary.

Maximum Likelihood Estimation

Suppose you have n samples drawn $\{x_i\}_{i=1}^n$ i.i.d from a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. What is the negative log-likelihood (NLL) of these samples? Derive the MLE (Maximum Likelihood Estimator) for μ and σ using these samples. Comment on the behavior of your estimators when $n \to \infty$. see notes here http://jrmeyer.github.io/machinelearning/2017/08/18/mle.html This function is twice differentiable so you can take double derivative and see the conditions for convexity, strong convexity.

Linear Regression

Let's say you have 3 points from a 3 dimensional space, namely $x_1 = [2, 1, 0], x_2 = [1, 1, 1], x_3 = [0, 1, 0]$. Also let, $y_1 = 0, y_2 = 5, y_3 = 2$. Assume there is some true $w \in \mathbb{R}^3$ such that $w^T x_i = y_i$.

1. Write expression for closed form solution for w. Does it exist? If yes, calculate it.

$w = X^{-1}y = [1, 2, 4]$ because X is invertible.
If it was not the case but X^TX was invertible, then
$w = (X^TX)^{-1}Xy = [1, 2, 4]$

2. Do at least 2 steps of gradient descent (manually). Use learning rate 0.02 and initialize w to $[0, 1, 0]$. At each step compare the current estimate by gradient descent to the closed form solution. $L(w) = \frac{1}{2}\|Xw - y\|_2^2$. We want to minimize this function using gradient descent. Using the above information, we get the following iterates which are converging towards the optimal solution:

$[-0.20, 1.40, 0.40]$
$[-0.14, 1.58, 0.70]$
$[-0.02, 1.69, 0.95]$
$[0.09, 1.78, 1.19]$
$[0.19, 1.85, 1.40]$
$[0.29, 1.91, 1.59]$

Probability

Your friend has a biased coin i.e. probability of head is $\theta \neq 1/2$. You agreed to play a game with your friend. Your friend tosses the coin and you need to guess whats on the outcome of the toss. You are overly confident about your predictions so you agreed to the conditions that you will pay $1 if you are wrong otherwise nobody pays anything.

1. Without knowing θ, what will be your strategy? What will be the expected amount of money you will be paying to your friend.

In this case, the best strategy would be to predict randomly, i.e., probability of head and tail $= \frac{1}{2}$. Denote your prediction by $\hat{x}$ and the true outcome as x. Then the expected loss,

$$E_{x,\hat{x}}[1(\hat{x} \neq x)] = \Pr(x = H \wedge \hat{x} = T) + \Pr(x = T \wedge \hat{x} = T) = \frac{\theta}{2} + \frac{1-\theta}{2} = \frac{1}{2}$$

2. Can you estimate θ while playing the game? What will be the most likely estimate of θ after n rounds of play. What happens to

your estimate if you keep playing the game forever. (You might go bankrupt though)

It will be a standard setting of MLE estimation for Bernoulli distribution. We can denote $x = 1$ when it is head and $x = 0$ otherwise, and do MLE estimation for distribution $p_\theta(x) = \theta^x(1-\theta)^{1-x}$ using samples $\{x_i\}_{i=1}^n$. The estimate will be $\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^n x_i$. If we keep playing this game forever, i.e., let $n \to \infty$, then by the law of large numbers $\hat{\theta}_n \to \theta$.

3. Suppose your friend's friend told you the θ, what will be your strategy? If you use this strategy what will be your expected loss?

You have two options that can minimize your loss. Either predict $\hat{x}$ with $p_\theta(x)$, this will have an expected loss $= 2\theta(1-\theta)$ or predict $\hat{x} = T$ if $\theta < \frac{1}{2}$ otherwise predict $\hat{x} = H$. This strategy will have an expected loss $(1 - \max\{\theta, 1-\theta\})$. The second one turns out to be the optimal one after comparing both of them for $\theta > \frac{1}{2}$ and $\theta < \frac{1}{2}$ cases.

4. By now may be you know you are loosing a lot of money without knowing θ. Instead of telling the true θ, your friend's friend told you that the true θ follows a Beta distribution with parameters $\{\alpha, \beta\}$. How will you use this prior knowledge to improve your estimation? Derive the estimator using this prior knowledge and n rounds of play.

See these slides for detailed analysis http://www.mi.fu-berlin.de/wiki/pub/ABI/Genomics12/MLvsMAP.pdf