COMS 4771 002 (Fall 2017) Exam 1 October 19, 2017

Please show all your work! Answers without supporting work will not be given credit.
Write your answers in spaces provided.
There are total thirteen pages including two blank pages at the end for scratch work.
You have 1 hour and 15 minutes to complete this exam.
You may use any result from the lectures or the homeworks without proof.

Name & UNI:____

1. **(16 points)** For each statement below state either True or False. Justify your answer by providing a short explanation/proof sketch (for true statements) or a counter example (for false statements).

   (a) __________ Bayes classifier (i.e. the optimal classifier) always yields zero classification error.

   (b) __________ Kernel regression with a box kernel typically yields a discontinuous predictor.

   (c) __________ If we train a Naïve Bayes classifier using infinite training data that satisfies all of its modeling assumptions (e.g., features in the training data are independent given the class labels), then it will achieve zero training error over these training examples.

(d) ____________ Suppose you are given a dataset of cellular images from patients with and without cancer. If you are required to train a classifier that predicts the probability that the patient has cancer, you would prefer to use Decision trees over Logistic regression.

(e) ____________ Given a $d \times d$ symmetric positive semidefinite matrix A. The transformation $\phi : x \mapsto x^{\mathsf{T}} A x$ is a linear transformation.

(f) ____________ Ordinary Least Squares (OLS) regression is consistent. That is, as the number of samples approaches infinity, OLS error approaches the error of the optimal regressor.

(g) ____________ Consider the constraint optimization problem: $\left\{ \begin{array}{rl} \min_x & f_0(x) \\ \text{such that} & f_i(x) \leq 0, \quad 1 \leq i \leq n \end{array} \right.$.

This optimization problem is equivalent to the following *unconstraint* optimization problem: $\min_x f_0(x) + \sum_{i=1}^{n} I(f_i(x))$, where $I(\cdot)$ is a 'step function' defined as $I(u) = \left\{ \begin{array}{ll} 0 & \text{if } u \leq 0 \\ \infty & \text{otherwise} \end{array} \right.$.

That is, the optimal solution to the given constraint optimization problem is equal to the optimal solution for the given unconstraint optimization problem.

(h) ____________ For any $d \geq 1$, the unit L_0-ball in $\mathbb{R}^d$, that is, the set $\{x \in \mathbb{R}^d : \|x\|_0 \leq 1\}$, is *not* a convex set.

2. [**Maximum Likelihood Estimation and beyond (13 points)**] Your friend gives you a coin with bias b (that is, tossing the coin turns '1' with probability b, and turns '0' with probability $1 - b$). You make n independent tosses and get the observation sequence $x_1, \ldots, x_n \in \{0, 1\}$.

   (a) You want to estimate the coin's bias. What is the Maximum Likelihood Estimate (MLE) $\hat{b}$ given the observations $x_1, \ldots, x_n$?

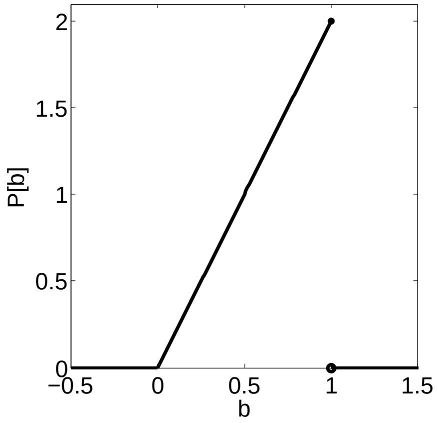
   (b) Is your estimate from part (a) an unbiased estimator of b? Justify your answer.

   (c) Derive a simple expression for the variance of this coin?

Cont.

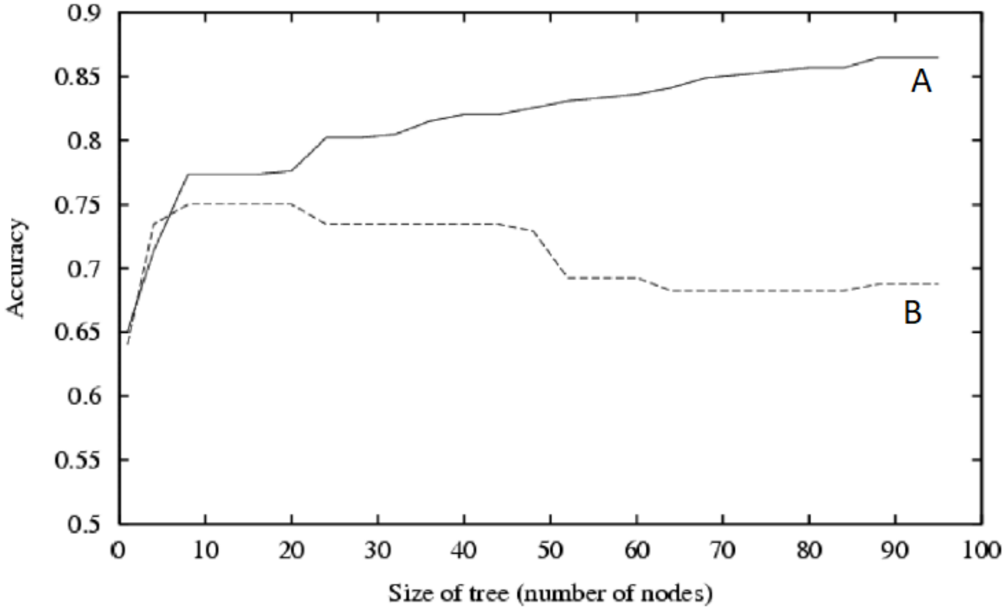
(d) What is the MLE for the coin's variance?

(e) Your friend reveals to you that the coin was minted from a faulty press that biased it towards 1. Suppose the model for the faulty bias is given by the following distribution:



Having this extra knowledge, what is the best estimate for the coin's bias b given the observation sequence? That is, compute: $\arg\max_b P[b \mid x_1, \ldots, x_n]$.

3. [**Analyzing Classifier Performance (6 points)**] Consider the plot below showing training and test set accuracy curves for decision trees of different sizes, using the same set of training data to train each tree.



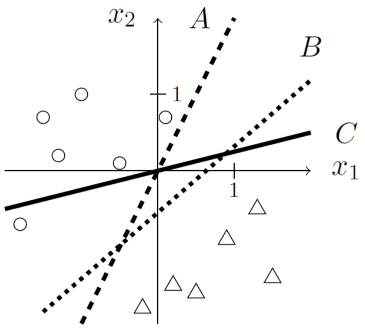
(a) What is the most likely correspondence between the training and test curves, and the curves marked 'A' and 'B' in the figure?

(b) Describe in one sentence how the training data curve will change if the number of training examples approaches infinity.

(c) How will the test data curve change under the same condition as in part ii (ie, for the case when number of *training* examples approaches infinity)?

4. **[Linear Classification]**

   (a) **(3 points)** The figure below depicts the decision boundaries of three linear classifiers (labeled A, B, and C) and the locations of labeled training data of n points (positive points are circles, negative points are triangles).



Consider the following algorithms for learning linear classifiers which could have produced the depicted classifiers given the depicted training data:

i. The Perceptron (making one pass over the training data, i.e., $T = n$).

ii. An arbitrary but fixed algorithm that minimizes training error for homogeneous linear classifiers.

iii. An algorithm that exactly solves the non-separable SVM optimization criterion. That is, SVM with slack variables optimization criterion.

What is the most likely correspondence between these algorithms and the depicted linear classifiers? Explain your matching to receive credit.

(b) **(12 points)** You are given a binary classification problem in $\mathbb{R}^2$ using features f_1 and f_2. For each of the following data preprocessing techniques, state and justify what happens to the training accuracy of any training error minimizing algorithm for homogeneous linear classification.

You should choose between the following options:

A. The preprocessing can only improve the performance.

B. The preprocessing absolutely does not change performance.

C. The preprocessing can only worsen the performance.

D. The preprocessing can either improve or worsen the performance.

i. Scaling the feature space: that is, apply the map $(f_1, f_2) \mapsto (cf_1, cf_2)$ (for some fixed constant $c > 0$ that is selected independently, without considering the training data)

ii. Translating the feature space: that is, apply the map $(f_1, f_2) \mapsto (f_1 + c, f_2 + c)$ (for some fixed constant $c \neq 0$ that is selected independently, without considering the training data)

iii. Augmenting the feature space: that is, apply the map $(f_1, f_2) \mapsto (f_1, f_2, c)$ (for some fixed constant $c \neq 0$ that is selected independently, without considering the training data)

iv. z-scoring the feature space: that is, apply the map $(f_1, f_2) \mapsto \left(\frac{f_1 - \mu_1}{\sigma_1}, \frac{f_2 - \mu_2}{\sigma_2}\right)$, where μ_i and σ_i are computed as the training data mean and standard deviation for the feature f_i respectively.

v. Combining the feature space: by applying the map $(f_1, f_2) \mapsto (f_1^2, f_1 f_2, f_2^2)$

vi. Combining the features: by applying the map $(f_1, f_2) \mapsto (f_1 + f_2)$

5. [**Nearest Neighbor Classification**]

   (a) **(5 points)** Two classifiers are said to be equal if on every test example they output the same prediction. Alice, Bob and Carol each construct a 3-NN classifier using the same training dataset S. They used the following distance metrics:

$$\begin{aligned} d_{\text{alice}}(x, x') &:= \|x - x'\| \\ d_{\text{bob}}(x, x') &:= 2e^{\|x-x'\|+1} \\ d_{\text{carol}}(x, x') &:= \|x - x'\|_1 \end{aligned}$$

   Whose classifiers would be equal for all non-empty training sets S? Justify your answer.

   (b) **(5 points)** Define $h_k(x; S)$ as the k-nearest neighbor classifier that takes in a training dataset S and a test example x and outputs a prediction $\hat{y} \in \{0, 1\}$. (so, $h_5(x, S_i)$ denotes the 5-NN classifier that is trained on S_i and predicts on x). Let S_1 and S_2 be two disjoint training datasets from the same underlying distribution, and consider the following statements:

   A. If $h_1(x, S_1) = 1$ and $h_1(x, S_2) = 1$, then $h_1(x, S_1 \cup S_2) = 1$.
   B. If $h_1(x, S_1) = 1$ and $h_1(x, S_2) = 1$, then $h_3(x, S_1 \cup S_2) = 1$.
   C. If $h_3(x, S_1) = 1$ and $h_3(x, S_2) = 1$, then $h_3(x, S_1 \cup S_2) = 1$.

   Which of the following statements is correct? (Justify your answer)

   i. Only A. is correct
   ii. Only B. is correct
   iii. Only C. is correct
   iv. Only A. and B. are correct
   v. Only A. and C. are correct
   vi. Only B. and C. are correct
   vii. All A. B. and C. are correct
   viii. None of the A. B. and C. are correct

(c) Just like the Perceptron or the SVM algorithm, you want to design a ‘Kernelized’ version of k-Nearest Neighbor classification. Since distance computations are at the core of any NN algorithm, you realize that in order to develop a successful ‘kernel’-NN algorithm, you need a way to efficiently compute distances in a (possibly infinite dimensional) Kernel space. Let $\Phi : \mathbb{R}^d \to \mathbb{R}^D$ denote an explicit mapping into a kernel space (where D can possibly be infinite).

i. **(5 points)** Show that Euclidean distances between data points in a kernel space can be written down as dot products, and thus can be computed efficiently for certain kernel transformations. That is, show that one can write:

$$\|\Phi(x) - \Phi(x')\|$$

just in terms of the dot products $\big(\Phi(x) \cdot \Phi(x')\big)$.

ii. **(5 points)** Write down the pseudo-code for Kernelized 1-Nearest Neighbor. Assume that you have access to the kernel function $K_\phi : (x, x') \mapsto \big(\phi(x) \cdot \phi(x')\big)$, that gives you the dot product in the feature space ϕ.

Kernel 1-Nearest Neighbor Algorithm

Inputs:
- Training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Kernel function: $K_\phi(\cdot, \cdot)$
- Test point: x_t

Output:
- Prediction on the test point: $\hat{y}_t$

Pseudo-code:

6. [**Regression (10 points)**] You want to design a regressor with good prediction accuracy. Knowing that various kinds of regularizations help, you decide to incorporate both the lasso and ridge penalties in the design of your optimization criterion. Given training data $(x_1, y_1), \ldots, (x_n, y_n)$ (where each $x_i \in \mathbb{R}^d$ and each $y_i \in \mathbb{R}$) you come up with the following optimization

$$\arg\min_w \ \left\|wX - y\right\|^2 + \lambda\Big[\alpha\|w\|_2^2 + (1-\alpha)\|w\|_1\Big],$$

where: (i) X is a $d \times n$ data matrix where the i^{th} column corresponds to training data x_i, (ii) y is a $1 \times n$ output vector where the i^{th} component corresponds to the training output y_i, and (iii) w is a $1 \times d$ parameter weight vector. $\lambda \geq 0$ and $\alpha \in [0, 1]$ are known trade-off parameters.

Show that this optimization can be turned into a lasso regression by appropriately augmenting the training data.

[blank page 1 for scatch work]

Cont.

[**blank page 2 for scatch work**]

The End.