True/False (75 points)

For each question, circle either True or False. If False, explain why.
5 points for True, 2 points for False, 3 points for explanation for False.

Point reductions for True False answers is based on the explanation (worth 3 points.) Saying True/False alone is worth 2 points. It is possible to get full credit for an explanation but no credit for the True/False alone. 1) There is an efficient (polynomial time) algorithm that can learn the optimal (smallest) decision tree that accurately captures the training data given 1000 examples of training data with 200 binary features and binary labels.
True False
Explanation if False:
False. Optimal learning is NP hard.

-2 a true statement that doesn't reference computational complexity

2) You are given two machine learning algorithms: A and B. Algorithm A yields a binary classifier with accuracy 90% on training data. Algorithm B yields a binary classifier with accuracy 70% on test data. You should use the binary classifier produced by Algorithm A.
True False
Explanation if False:
False. Depends on the test data.

3) Finding the optimal solution for linear regression can only be found through the use of a numerical convex optimization algorithm, such as gradient descent.
True False
Explanation if False:
False. There is a closed form (analytical solution.)

-2 something general about convex optimization
-2 simulated annealing
-2 brute force

4) You are given a convex function and a fixed step size η. Running stochastic gradient descent with step size η you will eventually converge to the optimal solution.
True False
Explanation if False:
False. We could oscillate around optimum.

True if answer says that a sufficiently small η was given.

5) Using maximum a posteriori (MAP) estimation for linear regression will return the same solution as minimizing the squared error.
True False
Explanation if False:
False. MLE is the same as minimizing error, not MAP.

-4 says true because in the limit they agree

6) Logistic regression cannot learn a non-linear decision boundary.
True False
Explanation if False:
True. It's a linear classifier.

Credit for False if explains a modification to logistic regression for non-linear learning.

7) The commonality between linear regression and logistic regression is that they both optimize the squared loss.
True False
Explanation if False:
False. Only linear regression uses squared loss.

-2 Says squared loss for linear regression but doesn't explain logistic regression correctly
-2 says they differ because logistic regression maximizes likelihood (both maximize likelihood)

8) The objective of logistic regression is to maximize the probability of the training data: $p(Y, X)$.
True False
Explanation if False:
False. Minimizes $p(y|x)$. -1 Says $p(x|y)$.

9) The Perceptron algorithm will always make a finite number of errors when trained on an infinite stream of linearly separable data.
True False
Explanation if False:
True. Several people said False, except if the problem is linearly separable,

which is exactly the case we are asking.

10) Online algorithms are fundamentally mistake driven algorithms, which means that they only update the current hypothesis when an error has been made in prediction.
True False
Explanation if False:
False. MIRA is margin based.

11) For linearly separable data sets, a Perceptron and SVM will learn the same parameters.
True False
Explanation if False:
False. Many possible hyperplanes for Perceptron.

-2 says that they differ in parameterization

12) The Kernel trick is effective since it can be used in both the primal and dual formulation of the SVM.
True False
Explanation if False:
False. Can only be used in the dual since it requires a dot product between

inputs.
-2 reasonable answer but doesn't say primal/dual
-1 references answer primal/dual but doesn't name them both
-1 reasonable answer but doesn't say it cannot be used in primal

13) As we increase the parameter K in kNN, we increase bias in the predictions, which means that we will obtain worse training accuracy.
True False
Explanation if False:
True. $k = 1$ **is perfect training accuracy.**

14) A feed forward artificial neural network with a single hidden layer containing a finite number of variables is a universal approximator among continuous functions on compact subsets of $\mathcal{R}^n$.
True False
Explanation if False:
False. Not true for a fixed number.

15) A maximum likelihood estimate is always inferior to a maximum a posteriori estimate (i.e., performs worse on training data) since maximum likelihood estimates underestimate the true variance.
True False
Explanation if False:
False. Sometimes it can do better; depends on the prior.

-2 says in the limit they meet

Short Answer (45 points)

16) (9 points) For each of the following problems, state which learning setting is the best fit: supervised learning, unsupervised learning, or reinforcement learning.

(a) Frank would like to build a computer game for checkers. One of the features of the game is that a user can play against the computer. Frank wants to ensure that the computer player is very good at checkers. He turns to machine learning to teach the computer how to play checkers.

Reinforcement learning

(b) Nancy works for a major search engine: Yahdu. She has access to millions of queries entered by users and which webpages the users visited after the search. Nancy wants Yahdu to return the webpages visited by users as the top answer for each query.

Supervised learning

(c) Deloris is working on a self-driving car. The car uses a camera to watch the road. Deloris is responsible for the steering component, which makes sure that the car stays on the road. To build this component, Deloris has access to thousands of images of the road taken from the car's camera and volunteers who have indicated where the road is located in each image.

Supervised learning

17) (10 points) State the name of the algorithm that optimizes the following loss functions.

(a) 0/1 Loss **Perceptron**

(b) Hinge Loss **SVM/MIRA**

(c) Squared error loss **Linear regression**

(d) Logistic loss **Logistic regression**

Which loss function is inappropriate for classification? Why?
Squared error is inappropriate for classification.
It is symmetric, goes to infinity on both ends, penalty is higher as we go away from the margin.
It is more suited for continuous values.
2 points for each right answer for parts (a) to (d), 1 point for squared loss. 1 point for explanation

18) (10 points) After years of searching, I found a biased coin! I called in my two friends, Mr. MLE and Ms. MAP. I then showed them my biased coin by flipping it 20 times and getting 15 heads and 5 tails. I claim it is biased.

(a) Does Mr. MLE (maximum likelihood estimation) think my coin is biased? Why?
Yes, MLE overfits the training data, under-estimates the variance
2.5 points for Yes, 2.5 points for the explanation

(b) Does Ms. MAP (maximum a posteriori estimation) think my coin is biased? Why?
No/Maybe, MAP uses prior so it depends on the prior.

2.5 points for Yes, 2.5 points for the explanation

Gave 2.5 points each for the explanation even if got Yes/No wrong.

19) (8 points) Suppose we have an SVM using each of the following kernels. Which of these kernels does not have an equivalent closed-formed SVM primal problem? Why?

(a) RBF Kernel: $K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||_2^2}{2\sigma^2}\right)$

(b) Polynomial Kernel: $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^T\mathbf{x}')^d$

(c) Linear Kernel: $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T\mathbf{x}'$

RBF, Polynomial kernels do not have a closed form.
RBF kernel is infinite dimensional in the primal. (ie, it cannot be written as a finite dot product. There are infinite terms in the Taylor series expansion of the exponential function)
Polynomial kernel needs transformation using a basis function in the primal.

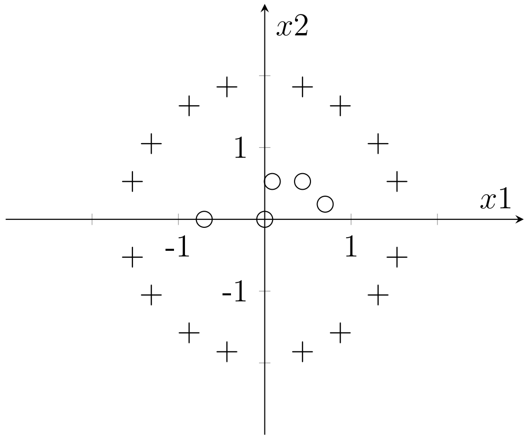
2 points for each right choice. 2 points for each explanation

20) (8 points) Suggest two modifications to the standard k-Nearest Neighbors algorithm that could alleviate over-fitting.

a) epsilon ball NN b) distance weighted kNN
4 points for each method. No points for PCA+kNN, kNN with higher k, etc. These are not modifications to the algorithm.

Long Answer (120 points)

21) (30 points) We decide to use logistic regression to fit the following training data.



Suppose a hypothesis class parameterized by **w** given by:

$$h(\mathbf{w}^T\mathbf{x}) = g(w_{(0)} + w_{(1)}x_{(1)} + w_{(2)}x_{(2)} + w_{(3)}x_{(1)}^2 + w_{(4)}x_{(2)}^2)$$

where $v_{(0)}$ is the 0th position of vector v, and

$$g(z) = \frac{1}{1+e^{-z}}$$

(a) Give values for the parameters **w** that correctly separates the given data.

(b) Write an equation in terms of vector **x** that represents this decision boundary.

(c) What is the purpose of $w_{(0)}$ in the hypothesis class?

(d) Suppose you are given a test data point of type ‘o’ at location $(0, 2)$. Would this point be correctly classified by your decision boundary?

One possible boundary:
a) w = [-1 0 0 1 1]
b) $x_1^2 + x_2^2 = 1$
c) w_0 **is the bias term**
d) No, (0,2) is outside the boundary. (For the boundary defined by (a))

7.5 points for each part.
(b) correct, (a) slightly off: -2 for (a)
(b) correct, (a) totally off: -5 for (a)
(a) slightly off and (b) off: -4 for (b)
(a) slightly off and (b) off, (d) correct: -4 for (d)
(a), (b) correct, (d) wrong, but right explanation: -2 for (d)
(a) correct, (d) wrong: -7.5 for (d)
(c) -5, -4, -3 based on the level of explanation
(a) partly right, (b) says sign(wx) : -5 for (b)

22) (30 points) Consider the difference between the SVM primal problem its dual. Suppose we have N training examples ($\mathbf{x}$,y), and each $\mathbf{x}$ is an M dimensional feature vector $\mathbf{x} \in \mathcal{R}^M$ with a binary label $y \in \{-1, 1\}$.

(a) How many parameters are used in the primary problem? What are they?

(b) How many parameters are used in the dual problem? What are they?

(c) One of the formulations (primal or dual) could have more parameters than the other. If this is the case, does one formulation have a greater likelihood of over fitting the data than the other?

(d) When considering a formulation for a linear SVM, which formulation would be prefer to ensure computational efficiency in terms of N and M?

(e) Given a training dataset $\{\mathbf{X}, \mathbf{Y}\}$, write the prediction rule for both the primal and dual for a test example $\mathbf{x}$.

(a) M (2): the weight vector (4)
(b) N (2): the dual variables associated with each example (4)
(c) No(2): the primal and dual problems are equivalent, and the primal and dual parameters are related by the data, i.e. primal parameters can be represented by the dual parameters and data(4)
(d) if $N > M$ choose primal (3); if $M > N$ choose dual(3)
(e) primal: $y = sign(w^T x)$ (3); dual: $y = sign(\sum_i \alpha_i y_i < x_i, x >)$ (3)

23) (30 points) Consider a generative model that takes example $\mathbf{x}$ containing M binary features and outputs a *vector* Y containing K elements, where each element k is binary label. For example, consider an image classification problem in which each element of Y is a different type of binary label, such as *"does this picture have a dog?"*, *"was this picture taken outside?"*, etc.

(a) Write the data likelihood for this model. How many parameters does it have in terms of the number of elements K and features M? Make no independence or conditional independence assumptions about the variables.

(b) Use the Naive Bayes assumption to assume that each element of $\mathbf{x}$ is conditionally independent given the entire vector Y. Write the data likelihood for this model. How many parameters are there in this model in terms of output elements K and input features M? Derive the maximum likelihood solution for these parameters.

(c) Assume that each element of Y is independent and that each element of $\mathbf{x}$ is conditionally independent given the entire vector Y. How many parameters are there in this model in terms of output elements K and input features M? Derive the maximum likelihood solution for these parameters.

(a) $P(x, Y) = P(x|Y)P(Y)$ **(6)**

-2 if did not facterize

-4 if only write down the conditional $P(x|Y)$ **or** $P(Y|x)P(x)$

$(2^M - 1) * 2^K + 2^k - 1$ **(4)**

-0 if no "-1"

-2 if correct for the likelihood you wrote

(b) $\prod_i P(x_i|Y)P(Y)M * 2^K + 2^k - 1$ **(8)**

-0 if no "-1"

-2 if $\prod_i P(x_i|Y)$ **and number of parameters is correct for the likelihood you wrote**

-2 if P is correct but the number of parameters is wrong

$P(x_i|Y) = \#(x_i, Y)/\#(Y)P(Y) = \#(Y)/N$ **(2)**

-1 if one of them is wrong

(c) $\prod_i P(x_i|Y)\prod_j P(Y_j)M * (2^k - 1) + K$ **(8)**

-0 if no "-1"

-2 if P is correct but the number of parameters is wrong

$P(x_i|Y) = \#(x_i, Y)/\#(Y)P(Y)_j = \#(Y_j)/N$ **(2)**

-1 if one of them is wrong

24) (30 points) A brand new computer game called DetectiveX is taking the world by storm. In this game, you play a character that moves around in a world solving mysteries. You play the game by taking turns. On each turn, your character can choose between several actions:

1. Move (4 actions): left, right, up, down. These actions move you around the 2-d map in the world.
2. Talk: talk to a nearby character
3. Take: take an item that is next to you. If you take an item, you place it in your bag. At all times, you know what items are currently in your bag.
4. Look: look closely at something nearby. When you look, you can gain information about the world (i.e., you see an item.) You always have access to the global map of the world and it becomes filled in as you move and look.

The goal of the game is to maximize your score (points), which are displayed in the top right of the screen at all times. After every move, you either receive points or lose points, the number of which depends on the action you took on your turn.

You decide you'd like to build a machine learning algorithm to play this game and, with luck, enter the algorithm in the world wide DetectiveX competition (for big money!) Luckily, you've taken CS 475 so you have some good ideas about how to build an algorithm. Also, you have obtained a special simulator for this game which can be used to train your model. The simulator allows you to instantiate a state of the game and then take an action (and see the resulting points awarded.) Since you can reload the simulator to any state, you can go back and forth in the game to consider the consequences of each action.

Design a Perceptron algorithm that learns to play DetectiveX. The goal will be to take actions within the game that achieves the highest score. Make sure to include the following details:

1. The pseudo-code of the algorithm. Describe how you train the model and how you would use it during the competition.
2. The parameters you are learning and the update equation for the parameters.
3. The loss function you will use.
4. Examples of features your model will use. Be sure that you provide an example of at least one feature for every one of the 7 actions. The same feature can be used for multiple actions, and you need only give one example for each category. We aren't looking for an exhaustive list of features.

(Space to answer question)

structured perceptron solution:

(a) training (6):

- initialize $w = 0$
- for each state/action pair (x, y)
  - form the feature $f(x, y)$, where $f(x, y)$ is usually $M * K$ and all but the yth M elements are padded with 0, where here $K = 7$; see the slides for details
  - $\hat{y} = \arg\max_y w^T f(x, y)$;
  - if $\hat{y} \neq y$
    - update w as equation in (b)

competition or test (4):

- given w and current state,
- predict the action $\hat{y} = \arg\max_y w^T f(x, y)$;

(b)
- parameter vector w, which has the same dimension as $f(x, y)$ (3)
- update equation: $w = w + f(x, y) - f(x, \hat{y})$ (6)

(c) we use a loss function which is the largest increase of the score of a given a action from the score given the true action, i.e. $w^T(f(x, y) - f(x, hat(y)))$ (3)

- -2 if only give 0/1 loss or hinge loss

(d) any reasonable features, e.g.

- move: from a to b difference of number of people between place a and b (2)
- look: the surrounding light condition (2)
- take: the appearance of an object (2)
- talk: the identity of a person (2)

Note: it's ok to use classical binary perception and use 1-vs-all classification, but should specify how to handle the multi-class case (here is the actions). i.e. specify how to update different w_i

- so for (a) training, typically -4 points if didn't mention how to do 1-vs-all classification.
- for (b), typically -4 points if only give $w' = w + yx$ without subscript to index w and x for different action.

(Scratch space)

(Scratch space)