# CS273a Midterm Exam

Machine Learning & Data Mining: Fall 2012

**Thursday November 1st, 2012**

**Your name:**

**Name of the person in front of you (if any):**

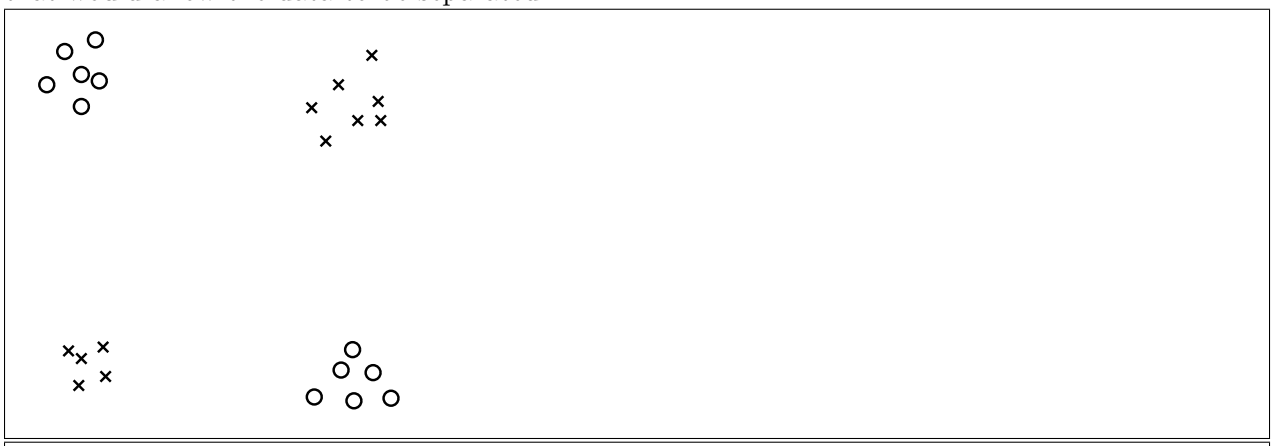
**Name of the person to your right (if any):**

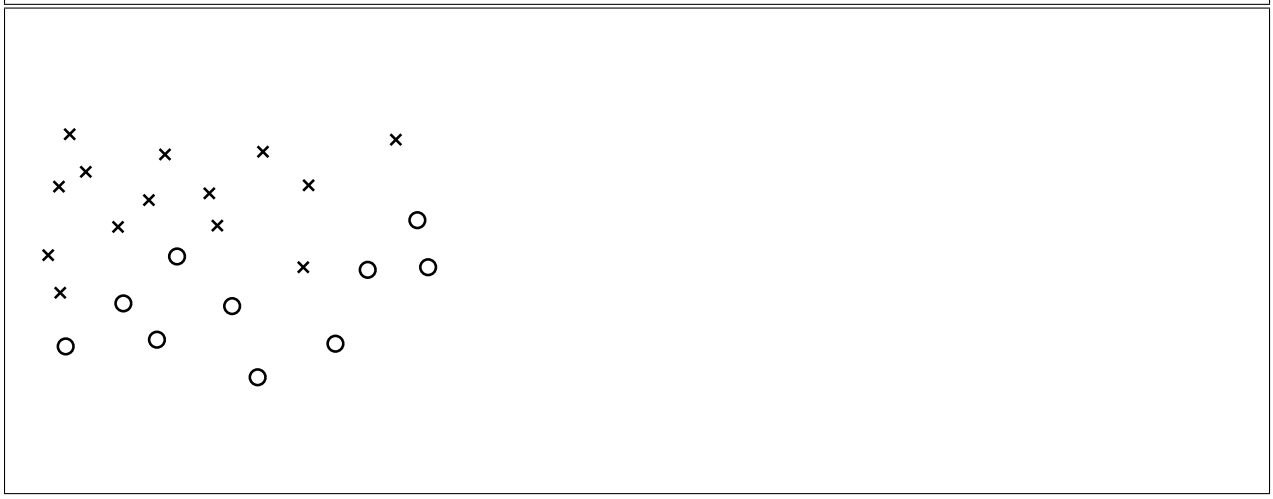
- Total time is 1:15. READ THE EXAM FIRST and organize your time; don't spend too long on any one problem.
- Please **write clearly** and **show all your work**.
- If you need clarification on a problem, please raise your hand and wait for the instructor to come over.
- Turn in any scratch paper with your exam.

(This page intentionally left blank)

## Problem 1: (12 points) Separability

For each of the following examples of training data, sketch a classification boundary that separates the data. State whether or not the data are linearly separable, and if not, give a set of features that would allow the data to be separated.





## Problem 2: (13 points) Under- and Over-fitting

(a) Suppose that we train a classifier, and discover that it achieves zero error. Are we likely to be over-fitting, under-fitting, neither, or do we need more information? Explain (1-2 sentences).

(b) **Circle one answer for each:**

Adding features to a linear classifier will make it **more** **equally** **less** likely to overfit the data.

Increasing the regularization parameter for a linear classifier will make it **more** **equally** **less** likely to overfit the data.

Increasing the step size in gradient descent for a linear classifier will make it **more** **equally** **less** likely to overfit the data.

Increasing the value of $k$ in a $k$-nearest neighbor classifier will make it **more** **equally** **less** likely to overfit the data.

Increasing the number of hidden nodes in a neural network will make it **more** **equally** **less** likely to overfit the data.

## Problem 3: (10 points) Regression

Suppose that we have training data $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$ and we wish to predict $y$ using the model:

$$\hat{y}(x) = a \log(x + b) + c$$

(a) Is this a linear or nonlinear regression model. Why?

(b) Write the mean squared error cost function for our predictor.

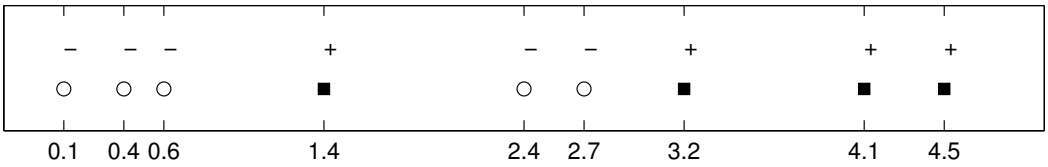
(c) Compute its gradient with respect to the parameters $a$, $b$, and $c$.

## Problem 4: (6 points) Optimization

(a) Give pseudocode for a stochastic (online or incremental) gradient descent algorithm to optimize the model in Problem 2. You do not need to have solved for the gradient $\nabla J(a, b, c)$ to do this; just assume it can be computed. Explain all the parameters used by your algorithm.

(b) Explain the difference between batch and stochastic gradient descent (1-3 sentences). Name one advantage for each.

**Problem 5: (12 points) Cross-validation and Nearest Neighbor**




Using the above data with one feature $x$ (whose values are given below each data point) and a class variable $y \in \{-1, +1\}$, with squares indicating $y = +1$ and circles $y = -1$ (the sign is also shown above each data point for redundancy), answer the following:

(a) Compute the leave-one-out cross-validation error of a 1-Nearest-Neighbor classifier. In the case of any ties, select the left-most neighbor at the same distance as the nearest.

(b) Compute the leave-one-out cross-validation error for a 3-Nearest-Neighbor classifier.

(c) Compute the leave-one-out cross-validation error for a 8-NN classifier. In the case of a tie, predict class +1.

## Problem 6: (14 points) VC Dimension

(a) Describe VC dimension in your own words, in a few (2-4) sentences.

(b) Give an example of a model in which the VC dimension is not equal to the number of parameters.

(c) Circle one answer for each:

Increasing the amount of training data in a linear classifier will **increase** **not change** **decrease** the VC dimension.

Increasing the number of features used in a linear classifier will **increase** **not change** **decrease** the VC dimension.

Increasing the regularization parameter for a linear classifier will **increase** **not change** **decrease** the VC dimension.

Exponentiating feature 1 before training (e.g., `x(:,1) = exp(x(:,1));`) a linear classifier will **increase** **not change** **decrease** the VC dimension.

## Problem 7: (12 points) Support Vector Machines

Consider a linear classifier, $T(wx^T + b)$, where $x = [x_1, \ldots, x_d]$ is a $d$-dimensional feature vector, $w = [w_1, \ldots, w_d]$ are the coefficients, and $b$ is the constant coefficient.

In class, I described how we could optimize a SVM written in constraint form,

$$\min_w \|w\|^2 \qquad\qquad s.t. y^{(i)}(wx^{(i)} + b) \geq 1$$

by optimizing $w$ along with a set of Lagrange multipliers $\alpha$:

$$J(w, \alpha) = \min_w \max_{\alpha \geq 0} \|w\|^2 + \sum_i \alpha_i \big(1 - y^{(i)}(w{x^{(i)}}^T + b)\big)$$

(a) By solving $\nabla_w J(w, \alpha) = 0$, show that the optimal value of $w$ is

$$w^* = \sum_i \alpha_i y^{(i)} x^{(i)}$$

(Hint: just take the derivative and solve for $w_1$ and argue symmetry.)

(b) For the following data, sketch the decision boundary, identify the support vectors, and give the value of $w$ and $b$ for a linear SVM trained on the data set.