CS273a Final Exam

Introduction to Machine Learning: Fall 2013

Thursday December 12th, 2013

Your name:

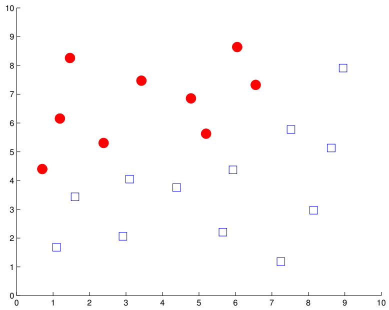
Your UCInetID (all caps):

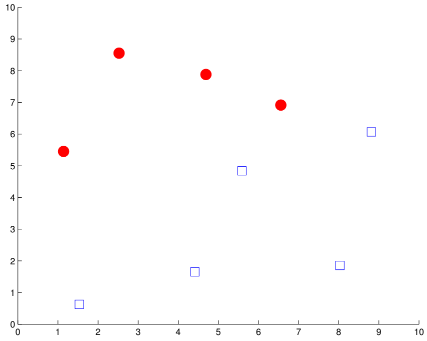
Your Seat (row and number):

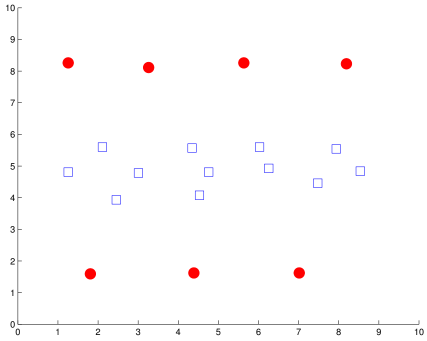
- Total time is 1:50. READ THE EXAM FIRST and organize your time; don't spend too long on any one problem.
- Closed book; one page of (your own) notes
- Please **write clearly** and **show all your work**.
- If you need clarification on a problem, please raise your hand and wait for the instructor to come over.
- Turn in any scratch paper with your exam.

Problem 1: 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:

Select the best choice to complete each statement.

Increasing the number of hidden nodes in a neural network will most likely
(**increase** **decrease** **not change**) the bias.

Decreasing regularization on the weights in logistic regression will most likely
(**increase** **decrease** **not change**) the VC dimension.

Increasing the amount of data will most likely
(**increase** **decrease** **not change**) the variance.

Increasing the regularization on a perceptron will most likely
(**increase** **decrease** **not change**) the bias.

Decreasing the maximum depth of a decision tree will most likely
(**increase** **decrease** **not change**) the VC dimension.

Increasing the depth of a decision tree will most likely
(**increase** **decrease** **not change**) the bias.

The predictions of a k-nearest neighbor classifier
(**will** **will not**) be affected by pre-processing to normalize the data.

Reducing the number of features using PCA will most likely
(**increase** **decrease** **not change**) the variance.

Linear regression
(**can** **cannot**) be solved using either matrix algebra or gradient descent.

The predictions of a regression tree
(**will** **will not**) be affected by pre-processing to normalize the features.

Problem 3: Regression

Suppose that we train a *non-linear* regression model on m data, where our prediction is

$$\hat{y}(x) = a + \exp(bx)$$

for two scalar parameters a, b.

(a) Write down the formula for the mean-squared error on the training data, and compute its gradient with respect to the parameters.

(b) Give pseudo-code for (batch) gradient descent on this problem. Be sure to specify initialization, the update itself (in enough detail to enable coding), and a stopping condition (again, in enough detail to enable coding).

(c) Give at least one advantage of batch gradient descent over stochastic gradient descent. In contrast, when would using stochastic gradient be more appropriate?

Problem 4: Naïve Bayes

Consider the following table of measured data:

x_1	x_2	x_3	y
1	1	1	0
1	1	0	0
0	1	0	0
1	0	1	0
1	1	1	1
0	1	1	1
0	0	1	1

We will use the three observed features x_1, x_2, x_3 to predict class y. In the case of a tie, we will prefer to predict class $y = 1$.

(a) Write down the probabilities necessary for a naïve Bayes (NB) classifier:

(b) Using your NB model, what value of y is predicted given observation $(x_1, x_2, x_3) = (000)$.

(c) Using your NB model, what is the probability $p(y = 1|x_1 = 1, x_2 = 1, x_3 = 1)$?

(d) Using your NB model, what is the probability $p(y = 1|x_1 = 0)$?

Problem 5: Perceptrons and VC Dimension

In this problem, consider the following perceptron model on two features:

$$\hat{y}(x) = \text{sign}(\, w_0 + w_1 x_1 + w_2 x_2 \,)$$

and answer the following questions about the decision boundary and the VC dimension.

(a) Describe (in words, with diagrams if desired) the decision boundaries that can be realized by this classifier

(b) What is its VC dimension?

Now suppose that I also enforce an additional condition on the parameters of the model: that **at most two** of the weights w_i are non-zero (so, at least one weight is zero).

(c) Describe (in words, with diagrams if desired) the decision boundaries that can be realized by this classifier

(d) What is its VC dimension?

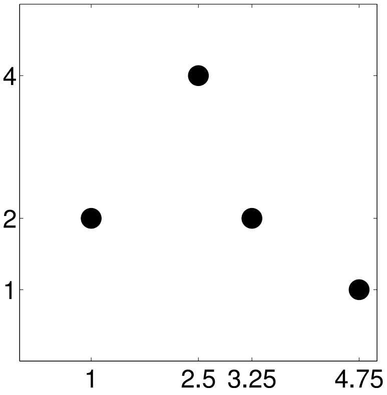
Finally, I enforce that **at most one** of the weights w_i is non-zero (so, at least two are zero).

(e) Describe (in words, with diagrams if desired) the decision boundaries that can be realized by this classifier

(f) What is its VC dimension?

Problem 6: Cross-validation

Consider a regression problem for predicting the following data points, using the k-nearest neighbor regression algorithm from class and the homework to minimize mean squared error (MSE). (Note: if you like, you may leave an arithmetic expression, e.g., leave values as "$(.6)^2$".)



(a) For $k = 1$, compute the training error on the provided data.

(b) For $k = 1$, compute the leave-one-out cross-validation error on the data.

(c) For $k = 3$, compute the training error on the provided data.

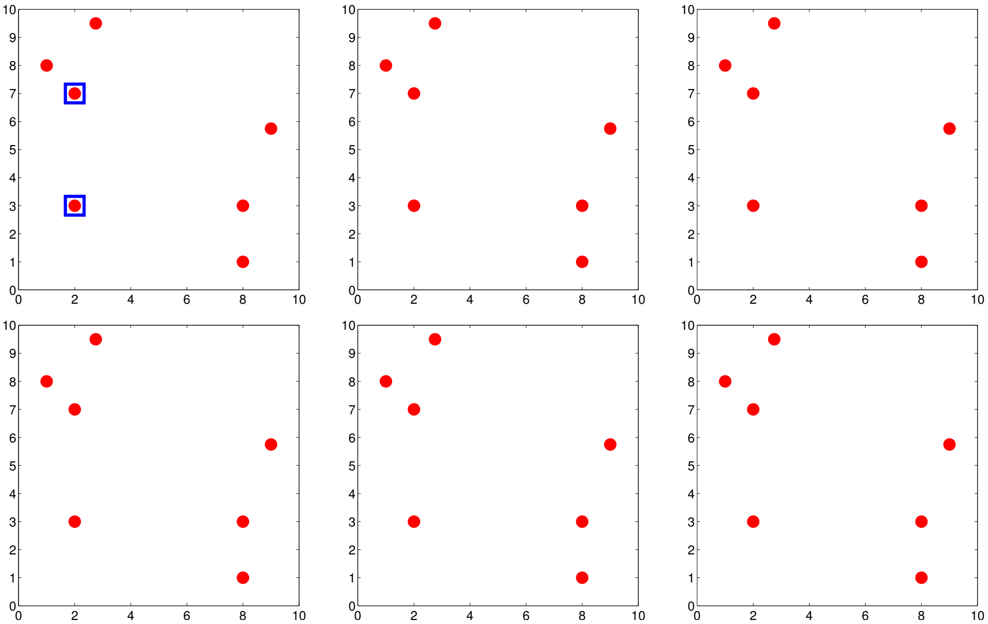
(d) For $k = 3$, compute the leave-one-out cross-validation error on the data.

Problem 7: Clustering

Consider the two-dimensional data points plotted in each panel. In this problem, we will cluster these data using two different algorithms, where each panel is used to show an iteration or step of the algorithm.

k-means

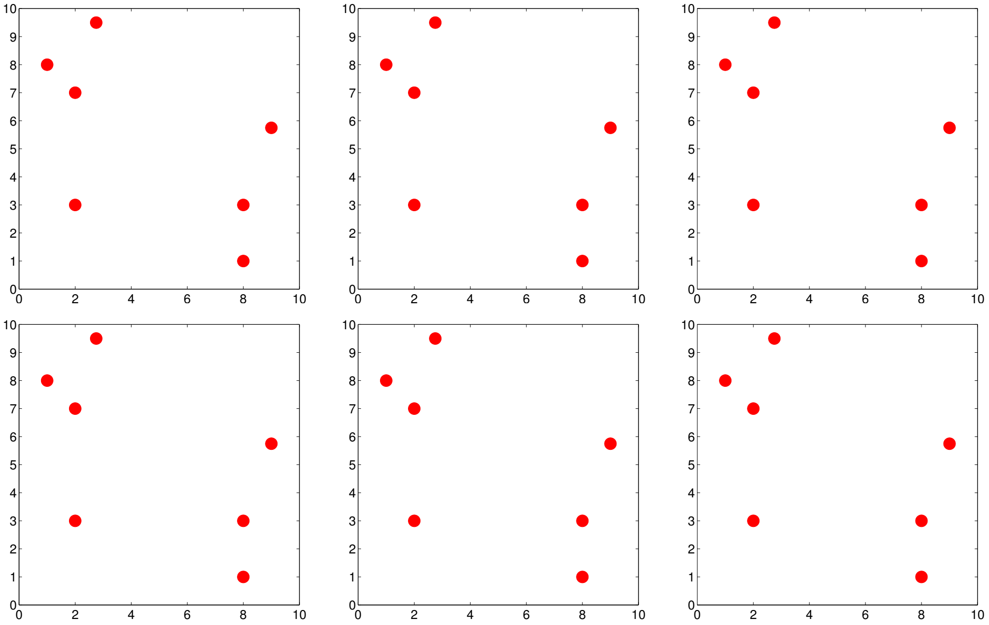
(a) Starting from the two cluster centers indicated by squares, perform k-means clustering on the data points. In each panel, indicate (somehow) the data assignment, and in the next panel show the new cluster centers. Stop when converged, or after 6 steps (3 iterations), whichever is first. It may be helpful to recall from our nearest-neighbor classifier that the set of points nearer to A than B is separated by a line.



(b) Write down the cost function optimized by the k-means algorithm, explaining your notation.

Linkage

(a) Now execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "complete linkage" (maximum distance) for the cluster scores. Stop when the algorithm would terminate, or after 6 steps, whichever is first. Show each step separately in a panel.



(b) What is the algorithmic (computational) complexity of the hierarchical clustering algorithm? Briefly justify your answer.