CS273 Final Exam

Introduction to Machine Learning: Winter 2015

Tuesday March 17th, 2015

Your name:

SOLUTIONS

Your UCINetID (e.g., myname@uci.edu):

Your seat (row and number):

- Total time is 1 hour 50 minutes. 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: (6 points) Multiple Choice

For the following questions, assume that we have m data points $y^{(i)}$, $x^{(i)}$, $i = 1 \ldots m$, each with n features, $x^{(i)} = [x_1^{(i)} \ \ldots \ x_n^{(i)}]$.

Circle one answer for each:

True or **false**: Early stopping can be used to reduce overfitting in neural networks.

True or **false**: The SVM learning algorithm will find the *globally optimal* model with respect to its objective function.

True or **false**: Increasing k in a k-nearest-neighbor classifier will decrease the bias.

True or **false**: Increasing the depth of a decision tree classifier will decrease the bias.

True or **false**: The VC dimension of a perceptron classifier is smaller than the VC dimension of a linear SVM.

True or **false**: If there exists a set of h instances that cannot be shattered by $f(x)$, then the VC dimension of f is less than h.

Problem 2: (4 points) Short Answer

Give one advantage of the dual (kernel) form of support vector machines over the primal (linear) form, **and** one advantage of the primal form over the dual.

Dual: may be easier to specify kernel similarity than features for linear classification
many kernels work in high / infinite dimensional feature spaces
better if # of features is very high (maybe)
∞ - d.m. kernel $\Rightarrow$ nonparametric predictor, often gets better as $m \to \infty$ (# data)

Primal: more efficient if $m \gg n$ (lots of data), in computation & model storage
Comp. & model storage are fixed (don't grow with m)
This also means it is often more efficient at test time (if $m \gg n$)
Easy to apply standard algorithms like SGD.

Problem 3: (12 points) Decision Trees

Consider the table of measured data given at right. We will use a decision tree to predict the outcome y using the three features, $x_1, \ldots, x_3$. In the case of ties, we prefer to use the feature with the smaller index (x_1 over x_2, etc.) and prefer to predict class 1 over class 0. You may find the following values useful (although you may also leave logs unexpanded):

$\log_2(1) = 0 \quad \log_2(2) = 1 \quad \log_2(3) = 1.59 \quad \log_2(4) = 2$

$\log_2(5) = 2.32 \quad \log_2(6) = 2.59 \quad \log_2(7) = 2.81 \quad \log_2(8) = 3$

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

(a) What is the entropy of y?

1 bit

(b) Which variable would you split first? Justify your answer.

X_1: 0 → $y=0$: 011, $y=1$: 001; 1 → $y=0$: 110 110, $y=1$: 100 111

X_2: 0 → $y=0$: X, $y=1$: 100 001; 1 → $y=0$: 110 110 011, $y=1$: 111

X_3: 0 → $y=0$: 110 110, $y=1$: 101; 1 → $y=0$: 011, $y=1$: 111 001

(c) What is the information gain of the variable you selected in part (b)?

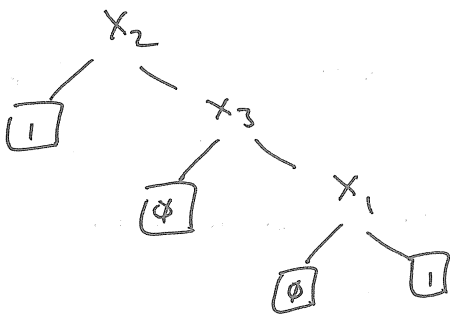
⇒ X_2.

$$IG = 1 - \left[\tfrac{1}{3} \cdot 0 + \tfrac{2}{3} \cdot \left(\tfrac{3}{4} \log \tfrac{4}{3} + \tfrac{1}{4} \log 4\right)\right]$$

$$= 1 - \tfrac{1}{2} \log \tfrac{4}{3} - \tfrac{1}{6} \log 4$$

$$= \tfrac{1}{2} \log 3 - \tfrac{1}{6} \log 4 \simeq .4591 \text{ bits}$$

(d) Draw the rest of the decision tree learned on these data.



X_3 next - gets 2 right (vs 1)

(can also calculate IG if desired, but clear by inspection)

Problem 4: (9 points) Gradient Descent & Latent Space Models

Suppose that, as in lecture, we wish to model a collection of text documents using a latent space model such as Latent Semantic Indexing. However, for interpretability, we would like our latent representation to be non-negative, so that each "direction" can only make a set of words more likely to appear, not less likely. As one solution, we use an exponential transform to ensure positive values, giving the model

$$x_j^{(i)} \approx \sum_k \exp(u_{ik}) \exp(v_{jk})$$

(so, data point i is a nonnegative linear combination u_i of nonnegative directions v_j). We wish to train our model (U, V) to minimize squared error from the obeserved data, and will train it using gradient descent.

(a) Write down an expression for J_{ij}, the mean squared error (MSE) of our model for element $x_j^{(i)}$, and for J, the overall MSE of our model.

$$J_{ij} = \left(x_j^i - \sum_k \exp(u_{ik}) \exp(v_{jk})\right)^2$$

$$J = \frac{1}{m} \sum_{ij} J_{ij} \qquad \text{(also OK to normalize by } \frac{1}{mn}\text{)}$$

(b) Compute the derivatives of J_{ij} with respect to u_{ik} and v_{jk}.

$$\frac{\partial J_{ij}}{\partial u_{ik}} = 2\left(x_j^i - \sum_k \exp(u_{ik}) \exp(v_{jk})\right) \cdot \left(-\exp(u_{ik}) \exp(v_{jk})\right)$$

(same!)

$$\frac{\partial J_{ij}}{\partial v_{jk}} = 2\left(x_j^i - \sum_k \exp(u_{ik}) \exp(v_{jk})\right) \cdot \left(-\exp(u_{ik}) \exp(v_{jk})\right)$$

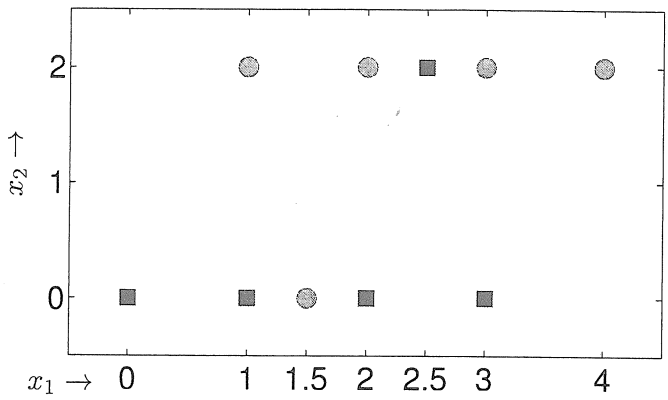
(c) Briefly describe how we could apply (e.g., give pseudocode for) stochastic gradient descent to learn u and v.

Init u, v to something (random usually)
Set stopping condition (eg # iterations) & step size α.
while (! done)
  for each i in random order
    for each j in random order (or do all j, depending on interpretation)
      $\hat{u} = u_i$, $\hat{v} = v$
      for $k = 1..K$
        $u_{ik} = u_{ik} - \alpha \frac{\partial J}{\partial u_{ik}}$
        $v_{jk} = v_{jk} - \alpha \frac{\partial J}{\partial v_{jk}}$ (use $\hat{u}, \hat{v}$ in ∂J to avoid overwriting)

Problem 5: (8 points) Cross-validation

Suppose that we learn a classifier on the following binary classification data. There are two real-valued features, x_1 and x_2, and a binary class $y \in \{0, 1\}$.

x_1	x_2	y
0	0	0
1	0	0
1.5	0	1
2	0	0
3	0	0
1	2	1
2	2	1
2.5	2	0
3	2	1
4	2	1



We decide to learn a decision tree as described in class. As in class, when the decision tree splits on the real-valued features, it puts the split threshold halfway between the data points on either side of the highest-scoring split. For example, if we first split on x_2, the algorithm would choose to split at $x_2 = 1$, which is halfway between the data at $x_2 = 0$ and $x_2 = 2$. In the case of ties, we prefer to predict class 0.

(a) What is the training error rate of a decision *stump* (decision tree with max depth 1, or two leaf nodes) trained on these data? split on $x_2 = 1$

$\Rightarrow$ 2/10 = 1/5 error

(b) What is the training error rate of a full decision tree (no maximum depth or pruning) trained on these data? split until all correct

$\Rightarrow$ $\emptyset$ error

(c) What is the leave-one-out cross-validation error rate of a decision *stump* (decision tree with max depth 1) trained on these data? All XV's split on $x_2 = 1$

$\Rightarrow$ 2/10 = 1/5 error

(d) What is the leave-one-out cross-validation error rate of a full decision tree (no maximum depth) trained on these data? All XVs split on $x_2 = 1$, Then at midpoint:

$\Rightarrow$ wrong at $x_2 = 0$, $x_1 = 1$, 1.5, 2

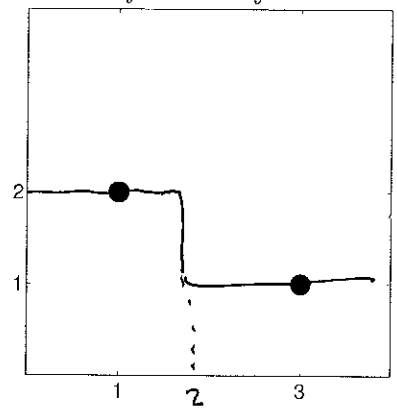
$x_2 = 2$, $x_1 = 2$, 2.5, 3

$\Rightarrow$ 6/10 = 3/5 error.

Problem 6: (8 points) Bagging

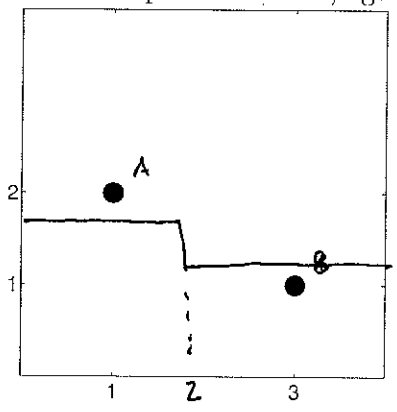
Consider a data set, consisting of two data points, plotted in each part.

(a) Draw the regression function (predicted values for all x) using a nearest-neighbor regressor. Label any necessary values on your graph.



$f(x) = \begin{cases} 2 & x < 2 \\ 1 & \text{ow.} \end{cases}$

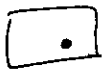
(b) Suppose that we create a very large ensemble of *bagged* nearest-neighbor regressors, using data set draws of size $m = 2$ during the boostrap sampling. Compute the regression function of the complete ensemble, again labeling any necessary values.



Bootstrap sampling draws data sets with replacement, which means we will get one of three possible sets, with probabilities:

$D = \{A, A\}$ — 1/4 of bootstraps ⇒ predict 2

$D = \{A, B\}$ — 1/2 of bootstraps ⇒ predict as in (a).



$D = \{B, B\}$ — 1/4 of bootstraps ⇒ predict 1

The bagged ensemble averages the predictions of its members ⇒ $1/4 \cdot 2 + 1/4 \cdot 1 + 1/2 \cdot \begin{cases} 2 & x < 2 \\ 1 & \text{ow.} \end{cases}$

$= \begin{cases} 1\,3/4 & x < 2 \\ 1\,1/4 & \text{ow.} \end{cases}$

(c) How does the training error in model (b) compare to the training error in (a)? (Note: you don't need to have answered (b) to answer this part.)

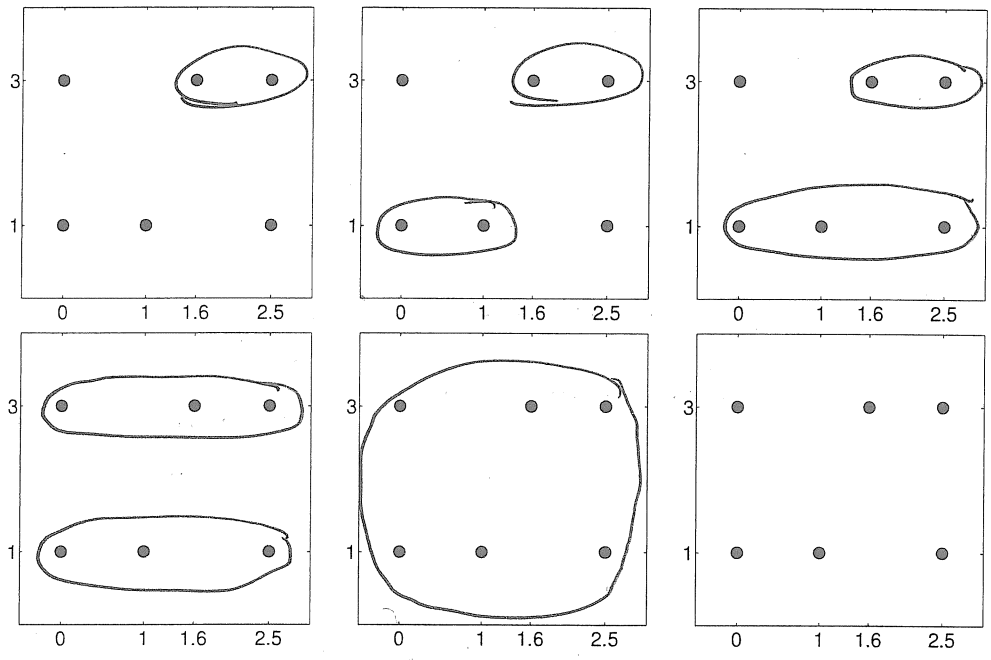
Training error has increased (MSE was 0, now $(1/4)^2$)

- our bagged model is overfitting less than the original; it is unable to "memorize" the data points.

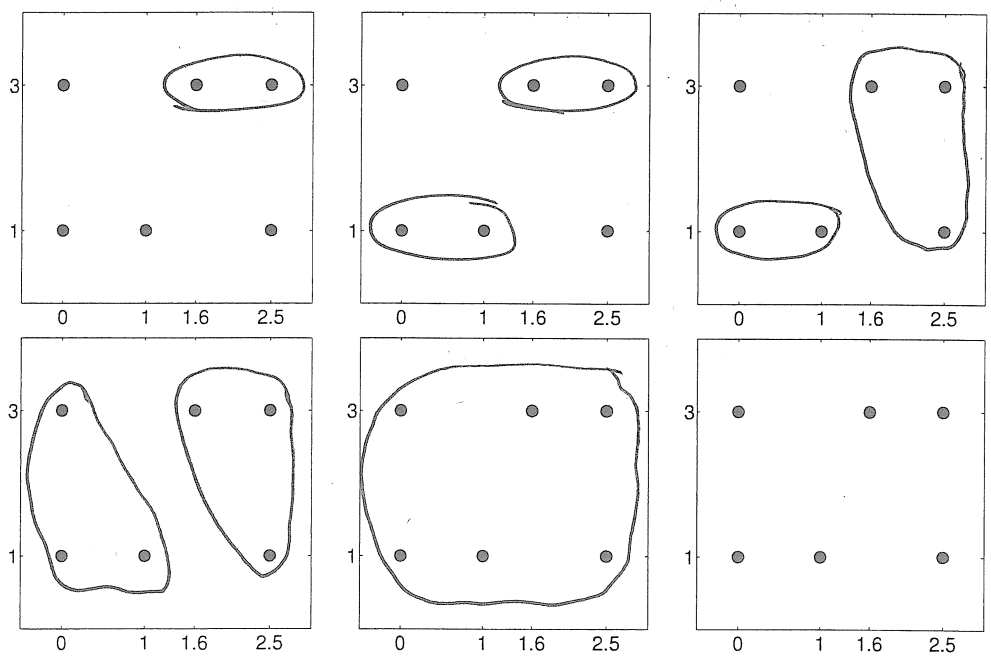
Problem 7: (8 points) Clustering

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.

(a) Execute the hierarchical agglomerative clustering (linkage) algorithm on these data points, using "single linkage" (minimum distance) for the cluster scores. Stop when converged, or after 6 steps, whichever is first. Show each step separately in a panel.

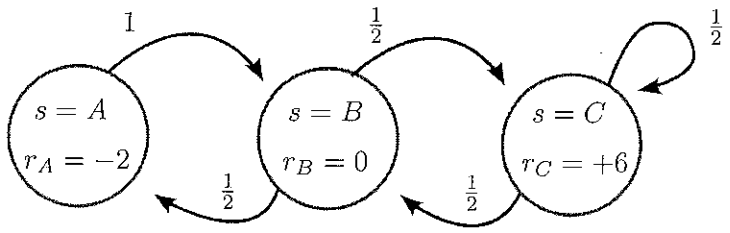


(b) Now execute hierarchical agglomerative clustering on the data points, but use "complete linkage" (maximum distance) for the cluster scores. Stop when converged, or after 6 steps, whichever is first. Show each step separately in a panel.



Problem 8: (8 points) Markov models

Consider the Markov model shown here:



$\Pr[A \to B] = 1.0$
$\Pr[B \to A] = \frac{1}{2}$
$\Pr[B \to C] = \frac{1}{2}$
$\Pr[C \to B] = \frac{1}{2}$
$\Pr[C \to C] = \frac{1}{2}$

where the transition probabilities are shown next to each arc and at right, and the rewards r_s associated with each state s are shown inside the circles. Assume a future discounting factor of $\gamma = \frac{1}{2}$.

(a) Compute $J^1(s)$, the expected discounted reward for state sequences of length 1 starting in each state s.

$J^1 =$

A	B	C
-2	0	6

(b) Compute $J^2(s)$, the expected discounted reward for state sequences of length 2 starting is each state s.

$J^2 =$

A	B	C
$-2 \cdot 1/2 \cdot 1 \cdot 0$	$0 + 1/2 \cdot 1/2 \cdot (-2)$	$6 + 1/2 \cdot 1/2 \cdot 0$
$= -2$	$+ 1/2 \cdot 1/2 \, (6)$	$+ 1/2 \cdot 1/2 \cdot 6$
	$= 1$	$= 7\,1/2.$

(c) Compute $J^3(s)$, the expected discounted reward for state sequences of length 3 starting is each state s.

$J^3 =$

A	B	C
$-2 + 1/2 \cdot 1 \, (1)$	$0 + 1/2 \cdot 1/2 \, (-2)$	$6 + 1/2 \cdot 1/2 \cdot (1)$
$= -1\,1/2$	$+ 1/2 \cdot 1/2 \, (7\,1/2)$	$+ 1/2 \cdot 1/2 \, (7\,1/2)$
	$= \frac{15}{8} - 1/2$	$= 6 + 1/4 + 15/8$
	$= 11/8$	$= 8\,1/8.$