CSE 546 Midterm Exam, Fall 2013

1. Personal info:
   - Name:
   - Student ID:
   - E-mail address:
2. There should be 16 numbered pages in this exam (including this cover sheet).
3. You can use any material you brought: any book, class notes, your print outs of class materials that are on the class website, including my annotated slides and relevant readings. You cannot use materials brought by other students. Laptops, PDAs, phones and Internet access are not allowed.
4. If you need more room to work out your answer to a question, use the back of the page and clearly mark on the front of the page if we are to look at what's on the back.
5. Work efficiently. Some questions are easier, some more difficult. Be sure to give yourself time to answer all of the easy ones, and avoid getting bogged down in the more difficult ones before you have answered the easier ones.
6. You have 80 minutes.
7. Good luck!

Question	Topic	Max score	Score
1	True/False	20	
2	Short Answer	20	
3	Decision Trees	8	
4	Linear Regression	12	
5	Logistic Regression	14 + 6 extra	
6	Boosting	14	
7	MLE	12 + 4 extra	
Total		100 + 10 extra	

1 [20 points, 2 points each] True/False (Explain in at most 2 sentences)

1. **true/false** Using a model with less bias is always better than using a model with more bias. Explain.

2. **true/false** Variance of a model typically decreases as the number of features increases. Explain.

3. **true/false** With the correct step size, gradient descent always converges to the optimum of the objective function for linear regression if the optimum exists. Explain.

4. **true/false** Making predictions with locally weighted least squares requires significantly more computation than making predictions with ordinary least squares. Explain.

5. **true/false** To predict the probability of an event, one would prefer a regression model trained with squared error to a classifier trained with logistic regression.

6. **true/false** Consider a model trained with Lasso. Adding the "debiasing step" to improve prediction quality generally helps more when the chosen regularization parameter λ is large rather than small.

7. **true/false** A good criteria for stopping when learning decision trees is to stop when the information gain is smaller than some value ϵ. Explain.

8. **true/false** In boosting, you can stop training weak classifiers if the error rate of the combined classifier is 0 on the original training data. Explain.

9. **true/false** As the amount of data increases, the true error of 1-NN approaches 0, assuming noise-free data. Explain.

10. **true/false** The perceptron algorithm is guaranteed to make a finite number of mistakes. Explain.

2 [20 points] Short Answer

1. [3 points] Consider using decision trees in a classification problem with 2 classes. What is the maximum training error that any dataset could possibly have? Explain with any relevant drawings and at most 4 sentences.

2. [3 points] Draw a dataset with 5 labeled examples for which LOOCV classification error on 1-NN is always 1 (1-NN gets it wrong every time). Use 2 classes. No explanation is needed.

3. [6 points] Suppose A, B, and C are binary attributes. Construct a dataset (by filling the table below) where the greedy algorithm we learned will not find the decision tree with minimum depth that achieves 0 training error. Construct a training set with no label noise, and show a minimum depth tree and the tree found by the greedy algorithm.

A	B	C	Class

4. [8 points] Consider the dataset in Figure 1. It has positive examples at

$$(2,0),(0,2),(1,1),(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}})$$

and negative examples at

$$(-2,0),(0,-2),(-1,-1),(-\frac{1}{\sqrt{2}},-\frac{1}{\sqrt{2}})$$

Recall that perceptron is trained on a sequence of examples. On each example, the weights are updated if perceptron makes a mistake in classifying that example.

Find a ordering of examples in this dataset on which perceptron makes at least 5 mistakes during training, or explain that no such sequence exists.

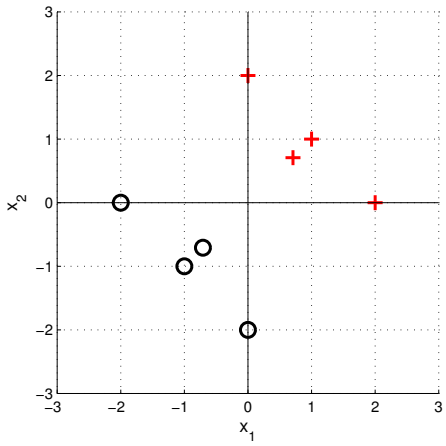


Figure 1: Dataset for the question on perceptron.

3 [8 points] Decision Trees

The following dataset will be used to learn a decision tree for predicting if people will get hired at a great company (Y) or not (N), based on their machine learning grade (High or Low), their GPA (High or Low) and on whether or not they did an internship during their PHD.

ML grade	GPA	Internship	(output) Hired?
L	H	Y	Y
L	L	N	N
L	L	Y	N
L	L	N	N
H	H	Y	Y
H	L	Y	Y
H	H	N	Y
H	L	N	Y

1. [1 point] What is the entropy H(Hired | Internship = N)? Briefly justify.

2. [1 point] What is the entropy H(Hired | GPA = H)? Briefly justify.

3. [6 points] Draw the full decision tree that would be learned for this data (assuming no pruning). You do not need to show the calculation of information gain.

4 [12 points] Linear Regression

Consider a regression problem in which we want to predict variable y from a single feature x. We have $n \geq 3$ data points, $(y_i, x_i)_{i=1}^n$. Consider two possible models to be estimated by ordinary linear regression,

$$y_i = w_0 + w_1 x_i + \epsilon_i \tag{1}$$

$$y_i = w_0 + w_1 x_i + w_2 x_i^2 + \epsilon_i \tag{2}$$

As in lecture, the error terms ϵ_i are independent and identically distributed from a normal distribution with zero mean.

1. [3 points] Derive a formula for estimating w_2 in Model 2. Assume w_0 and w_1 are known. Show your work.

2. [3 points] Will one model fit the **training** data better than the other, will they fit equally well, or is it impossible to say? Explain your reasoning.

3. [3 points] Will one model fit the **testing** data better than the other, will they fit equally well, or is it impossible to say? Explain your reasoning.

4. [3 points] Assume the true model is either Model 1 or 2. What technique regarding a change to the objective function of linear regression would be helpful in determining whether Model 1 or Model 2 is the true model (i.e., $w_2 = 0$)? Explain your reasoning.

5 [14 points + 6 extra credit] Logistic Regression

In this question we will consider logistic regression in two dimensions with various types of regularization. First, suppose we perform L2 regularization but the weights w_1 and w_2 are not necessarily penalized equally. That is, our objective function is now

$$F(\mathbf{w}, w_0) = \sum_i L(\mathbf{x}^i, y^i, \mathbf{w}, w_0) + \lambda_1 w_1^2 + \lambda_2 w_2^2$$

where $L(\mathbf{x}^i, y^i, \mathbf{w}, w_0)$ is the logistic loss function for example $(\mathbf{x}^i, y^i)$ and the remaining terms are regularization penalties.

Recall from class that $y^i \in \{0, 1\}$ (graphed as plus and circle, respectively) and

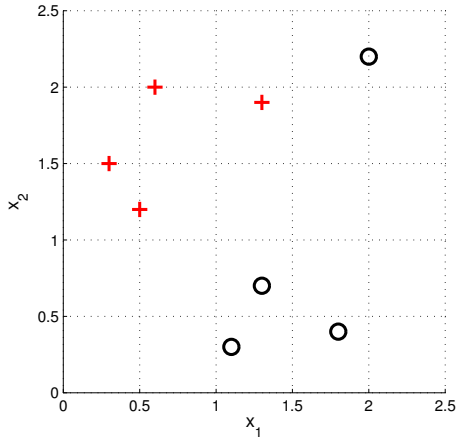
$$L(\mathbf{x}^i, y^i, \mathbf{w}, w_0) = y^i(w_0 + \mathbf{w}^T\mathbf{x}^i) + \ln(1 + \exp(w_0 + \mathbf{w}^T\mathbf{x}^i))$$

The graphs in the next page illustrate the data for this problem and will be used for recording your answers.

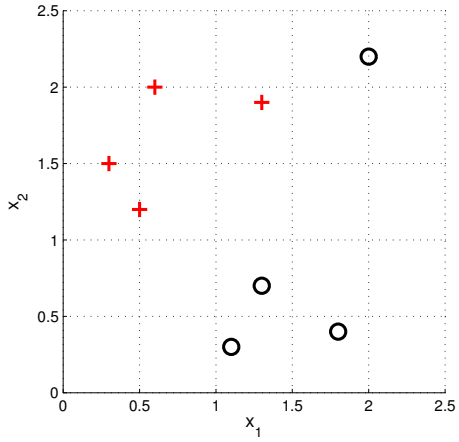
1. [2 points] Suppose λ_1 and λ_2 are both small but nonzero. In Figure 2(a), draw the decision boundary learned by logistic regression. No explanation is needed. (Note: for all these problems, your solution need not be exact. We are just looking for the correct points to be separated.)

2. [3 points] Now suppose λ_1 and λ_2 are both 0. Briefly explain (but do not draw) what happens to the decision boundary, the weights $\mathbf{w}$, and the value of $F(\mathbf{w}, w_0)$.

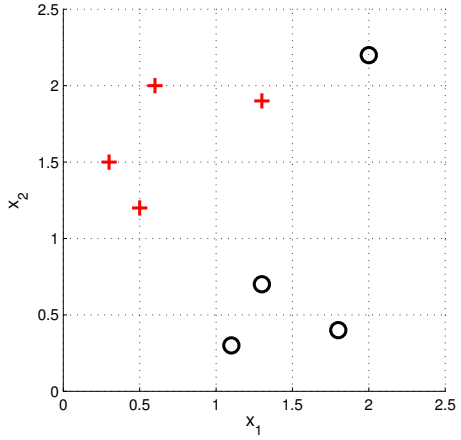
3. [3 points] Now suppose λ_1 is set to 0, but λ_2 is a very, very large value. Briefly explain what happens to the weights $\mathbf{w}$ and draw the resulting decision boundary in Figure 2(b).



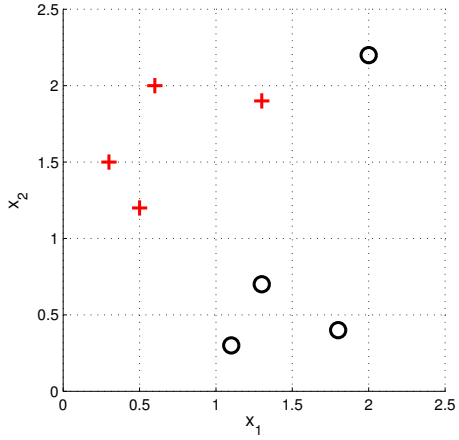
(a) Decision boundary for small λ_1 and λ_2



(b) Decision boundary for $\lambda_1 = 0$ and large λ_2



(c) Decision boundary for large λ_1 and $\lambda_2 = 0$



(d) Decision boundary for small λ_1 and λ_2 but large $\tilde{\lambda}$

Figure 2: Dataset for question 5 on logistic regression.

4. [3 points] Similarly, suppose λ_2 is set to 0, but now λ_1 is a very, very large value. Briefly explain what happens to the weights $\mathbf{w}$. Draw the resulting decision boundary in Figure 2(c).

5. [3 points] Now suppose that we are given additional prior knowledge about the weights $\mathbf{w}$. In addition to being small, we also believe $\mathbf{w}$ should be close to some given parameters $\tilde{\mathbf{w}}$ and $\tilde{w}_0$. In this case, these weights are

$$\tilde{w}_0 = 0, \tilde{w}_1 = -1, \tilde{w}_2 = 1$$

To ensure our estimate stays close to $\tilde{\mathbf{w}}$, we minimize the modified objective function $\tilde{F}(\mathbf{w}, w_0)$ with an added regularization term:

$$\tilde{F}(\mathbf{w}, w_0) = \sum_i L(\mathbf{x}^i, y^i, \mathbf{w}, w_0) + \lambda_1 w_1^2 + \lambda_2 w_2^2 + \tilde{\lambda}[\|\mathbf{w} - \tilde{\mathbf{w}}\|_2^2 + (w_0 - \tilde{w}_0)^2]$$

Assume the regularization parameters are chosen such that λ_1 and λ_2 are both quite small, but $\tilde{\lambda}$ is very, very large. Draw the resulting decision boundary in Figure 2(d). You do not need to explain.

The remaining problems in this section (5.6 through 5.8) are extra credit.

6. [2 points extra credit] The objective function $\tilde{F}(\mathbf{w}, w_0)$ is useful in online learning, where we have learned a weight vector $\hat{\mathbf{w}}^{(t)}$ from t previous examples but then encounter a new datapoint $(\mathbf{x}^{(t+1)}, y^{(t+1)})$. In order to to avoid storing previous examples in memory, we update $\hat{\mathbf{w}}^{(t)}$ by solving the following optimization problem with only the new datapoint and previous solution:

$$\hat{\mathbf{w}}^{(t+1)} = \arg\min_{\mathbf{w}} L(\mathbf{x}^{(t+1)}, y^{(t+1)}, \mathbf{w}) + \lambda \|\mathbf{w}\|_2^2 + \tilde{\lambda} \left\|\mathbf{w} - \hat{\mathbf{w}}^{(t)}\right\|_2^2$$

(For simplicity, we're now ignoring the offset term w_0.) This optimization problem is convex. Name a technique that solves this problem. Explain in a couple of sentences.

7. [2 points extra credit] Now assume in this case that $\tilde{\lambda}$ is large enough and L is smooth enough that

$$\nabla[L(\mathbf{x}^{(t+1)}, y^{(t+1)}, \hat{\mathbf{w}}^{(t+1)}) + \lambda \left\|\hat{\mathbf{w}}^{(t+1)}\right\|_2^2] \approx \nabla[L(\mathbf{x}^{(t+1)}, y^{(t+1)}, \hat{\mathbf{w}}^{(t)}) + \lambda \left\|\hat{\mathbf{w}}^{(t)}\right\|_2^2]$$

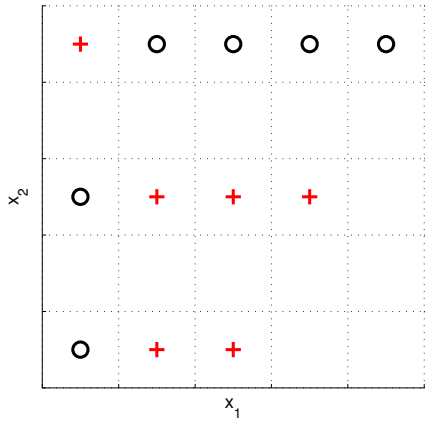
Using this assumption, solve for an approximate update rule for $\hat{\mathbf{w}}^{(t+1)}$ (by approximately solving the optimization problem above). Your result should be a closed-form solution involving $\nabla L(\mathbf{x}^{(t+1)}, y^{(t+1)}, \mathbf{w}^{(t)})$.

8. [2 points extra credit] In roughly two sentences, describe how the update rule you just derived above compares to the perceptron algorithm and stochastic gradient descent.

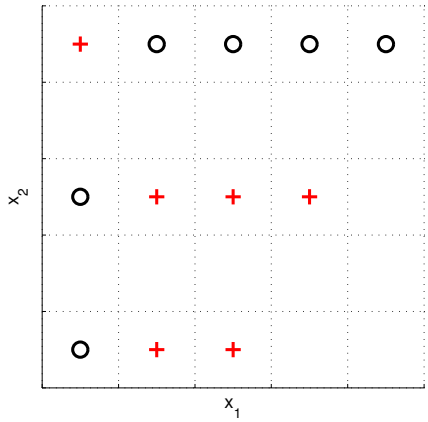
6 [14 points] Boosting

Consider the small dataset in Figure 3. We'd like to learn a classifier to separate pluses from circles using the AdaBoost algorithm. Each data point x_i has a class $y_i \in \{-1, +1\}$, where -1 corresponds to circle and $+1$ to plus.

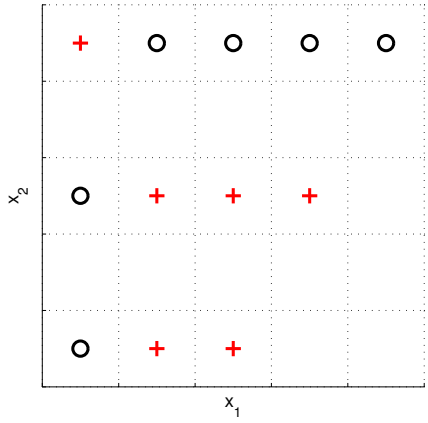
For this problem, we use weak learners with separating planes *parallel to a particular axis*, i.e., the decision boundary for each classifier is either a vertical or horizontal line. When training the new weak learner $h_t(x)$, we choose the split that maximizes the weighted classification accuracy with respect to current weights D_t (i.e., choose h_t that maximizes $\sum_i D_t(i)\delta(h_t(x_i) = y_i)$). Note that $h_t(x)$ only takes values in $\{-1, +1\}$, depending on whether it classifies x as a circle or plus respectively.



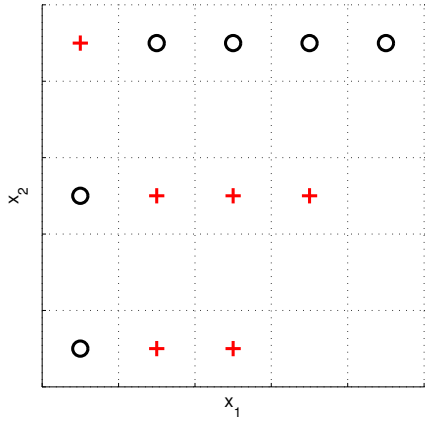
(a) Decision boundary after 1st iteration



(b) Decision boundary after 2nd iteration



(c) Example with *lowest* weight



(d) Example with *highest* weight

Figure 3: Dataset for question 6 on boosting.

1. [2 points] Draw the boosting algorithm's decision boundary after the first iteration (after the first weak learner is chosen) on Figure 3(a). Don't forget to mark which parts of the plane get classified as "+" and which as "∘". No explanation is needed.

2. [2 points] Now complete the second iteration of boosting. Draw the two decision boundaries after the first two iterations in Figure 3(b). Mark which parts of the plane are classified as "+" and "∘" by each classifier. No explanation is needed.

3. [5 points] In AdaBoost, we choose α_t as the weight of the t-th weak learner, where $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$ and $\varepsilon_t = P_{x \sim D_t}[h_t(x) \neq y]$ (i.e. ε_t is the weighted fraction of examples misclassified by the t-th weak learner). The update rule for the weight $D_t(i)$ of example i from step t to $t+1$ is

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalizing constant ensuring that the weights sum to 1 over examples. Which is larger: α_1 or α_2? Show your work.

Hint: $\sqrt{3} \approx 1.7$, $1/\sqrt{3} \approx 0.6$.

4. [1 point] After two iterations of boosting, how many training examples are misclassified? No explanation is needed.

5. [1 point] Using Figure 3(c), mark the training example(s) with *lowest* weight (D_t) after two iterations of boosting. No explanation is needed.

6. [1 point] Using Figure 3(d), mark the training example(s) with *highest* weight (D_t) after two iterations of boosting. No explanation is needed.

7. [2 points] For this dataset, will Boosting ever achieve zero training error? Explain in one or two sentences.

7 [12 points + 4 extra credit] Maximum Likelihood Estimation

Suppose we know a continuous random variable X is uniformly distributed between values 0 and a positive number c, but c is unknown. To help estimate c, we observe N independent samples $x_1, x_2, \ldots, x_N$ of X.

1. [2 points] Write the joint likelihood $P(x_1, x_2, \ldots, x_N \mid c)$.

2. [6 points] Find the maximum likelihood estimate of c. Your answer should be a closed-form solution for the estimate $\hat{c}$. Show your work.

3. [4 points] In this case, is the maximum likelihood biased or unbiased? Justify your answer.

The following question is extra credit.

4. [4 points extra credit] Show that, given enough samples N, the estimate $\hat{c}$ is within a small ratio ϵ from the true parameter c with high probability. That is, $\hat{c}$ satisfies

$$(1 - \epsilon)c \leq \hat{c} \leq (1 + \epsilon)c$$

with high probability. How many independent samples N are required in order to make this gaurantee with probability $1 - \delta$?