CSE 517a (Machine Learning): Midterm

Wednesday, March 3rd, 2010

.

First Name	
Last Name	
cec login	
Question 1	/
Question 2	/ 16
Question 3	/ 20
Question 4	/ 22
Question 5	/ 20
TOTAL	/

This page was unintentionally left blank. (Seriously, how do you get rid of it?)

1. Short questions

1 Point for correct answer, +2 Points for explanation (if applicable)
In case of "false" provide a small justification of your answer.

T/F A classifier trained on less training data is less likely to overfit.

F. A small training data set is likely to differ from the true distribution of the real data, causing the classifier to overfit.

T/F As the amount of training data increases, the training error goes down.

F. As the size of training data increase, it becomes harder for the model to classify all of them correctly.

T/F The best machine learning algorithms make no assumptions.

F.

T/F The regularization constant should be chosen on the test data set.

F. It should be chosen on the validation set.

T/F In AdaBoost, weights of the misclassified examples go up by the same multiplicative factor per iteration.

F. The multiplicative factor depends on the classification error at each iteration.

T/F The training error of Adaboost decreases exponentially.

T.

T/F Linear models are too restricted to overfit.

F. It can still overfit in high-dimensional feature space.

T/F Boosting is used to reduce the bias of low-variance classifiers.

T.

T/F Minimizing the log-loss for linear classifiers is equivalent to choosing the weight vector w with the Maximum Likelihood Estimate for the conditional probability $P(y|x;w) = \frac{1}{1+e^{-w^\top xy}}$ and $y \in \{+1, -1\}$.

T.

T/F The zero-one loss can be minimized with the gradient descent algorithm.

F. The gradient of the loss function is always 0 except at the origin. Gradient doesn't leads the optimization solver to the minimum of the function.

2. Nearest Neighbor Classification

[16 Points]

1. How does the *bias* of $k-NN$ change as k *increases*? [2]

   As k increases, the bias increases.

2. How does the *variance* of $k-NN$ change as k *decreases*? [2]

   As k decreases, the variance increase.

3. Joe implemented a function $predictions = knnclassify(xTr, xTe, k)$. He wants to know the leave-one-out training error of $1-$nearest neighbors and runs $predictions = knnclassify(xTr, xTr, 1)$. He is amazed and full of joy about how well his algorithm performs. What could have gone wrong? How would you fix it? [4]

   Leave-one-out cross validation: one data point from xTr should be used as cross-validation set each time and repeat until all data points in the training have been used as validation set. It is very possible Joe didn't exclude that point from xTr. Hence, the accuracy is very high.

4. How does the curse of dimensionality affect kNN classification of uniformly sampled data points within a $[0,1]^d$ space? [4]

   For uniformly sampled data points, assume the k nearest neighbors are within a smallest hyper-cube with edge-length l. Then we have:

   $$l^d = \frac{k}{n} \Rightarrow l = (\frac{k}{n})^{\frac{1}{d}} \xrightarrow{\text{as } d \text{ increase}} 1$$

   In other words, as the dimensionality increases you have to search a larger and larger space (in the limit the entire space). Consequently, no points are close. All data points become similarly far away from your query point.

5. Why does kNN still work on high-dimensional images of hand-written digits despite the curse of dimensionality? [2]

   The intrinsic dimensionality is low.

6. When would you use $KD-Trees$ over $Ball-trees$ and vice versa? [2]

   For very low dimensional data KD-Trees tend to perform better because they have lower overhead. Ball-tree's can still yield speed-ups with higher dimensional data because they can adapt better to low intrinsic dimensionality.

3. Decision Trees

[20 Points]

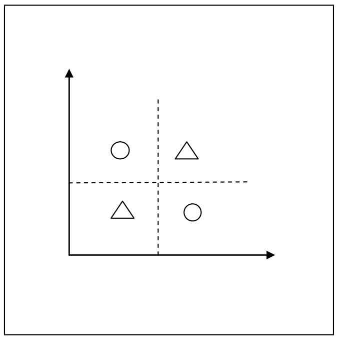
1. In the id3tree-algorithm what are the criteria to decide when to stop recursing and create a leaf? [4]

   (a) Can no longer split (all the data points have the same feature vector)

   (b) All the data points have the same label

   (c) Depth of tree reaches certain threshold

2. Why are decision trees called myopic? [2]

   
   It's a greedy algorithm and only looks at one feature at a time. In the example, the first split does NOT decrease the impurity measure, however, if combining two feature splits, we reduce the impurity measure to 0. If this data set had a third feature that has small correlation with the label, decision trees would always prefer this third feature over the perfect split combination of (feature 1 then feature 2).

3. If you build a single decision tree, how can you reduce the chance of overfitting (no need to state any algorithm)? [3]

   Limit the complexity of the tree. Use a validation data set to prune the number of leafs.

4. Assume you are given the following data set:

person	side-kick	ears	smokes	height	class
Batman	n	y	n	180	good
Robin	y	n	n	176	good
Alfred	n	n	n	185	good
Penguin	n	n	y	140	evil
Catwoman	n	y	n	170	evil
Joker	n	n	n	179	evil

5. (a) What is the expected class entropy after a binary split at *height* ≤ 160? (No need to compute the actual number, just write down the expression you could type into a calculator.) [3]

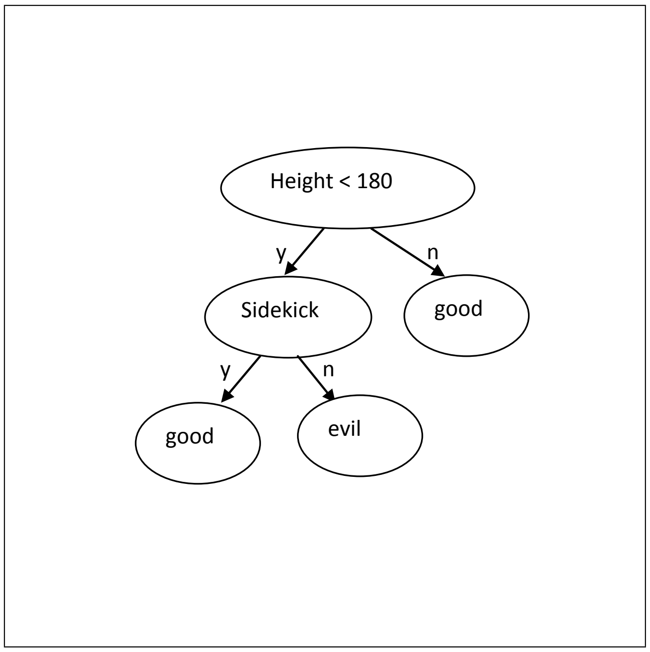
$$\frac{1}{6} \times 0 + \frac{5}{6} \times \left(\frac{3}{5} \cdot \log(\frac{5}{3}) + \frac{2}{5} \cdot \log(\frac{5}{2}) \right)$$

   (b) What is the first feature that the id3 algorithm would pick to split on (using expected entropy)? (No justification required.) [2]

   Height.

continued on next page...

(c) Draw a full decision tree that the id3 algorithm would learn. [4]



(d) Given the following validation data, what would your validation error be (in terms of misclassified examples)? [2]

person	side-kick	ears	smokes	height	class
Riddler	n	n	n	170	evil
Batgirl	y	n	n	150	good
Timo	n	n	n	60	good

The built tree will correctly classify Riddler and Batgirl, but misclassify Timo. Hence, the validation error is $\frac{1}{3}$

4. Bias, Variance

[22 Points]

1. Write down the definitions of (squared) bias, variance and noise. [6]

$\bar{h}(x)$: The expected hypothesis;
$\bar{y}(x)$: The conditional expectation of y given x;
$h_D(x)$: The hypothesis trained on training data D;
Squared bias: $E_x\left[(\bar{h}(x)-\bar{y}(x))^2\right]$
Variance: $E\left[(\bar{h}(x)-h_D(x))^2\right]$
Noise: $E\left[y(x)-\bar{y}(x))^2\right]$

2. Describe a method to detect if your classifier suffers from too much bias or too much variance. [6]

Increase your training data set, and plot out the train error and test error with respect to the size of the training data. If the gap between train error and test error decreases as you increase your training data, then your classifier has too much variance; on the other hand, if the train and test error do not change much during the process, and your train error is above desired rate, then too much bias.

3. Describe *bagging*, and explain what effect it has on bias / variance. [4]

Bootstrap M new data sets, $D_1, D_2, \cdots, D_M$ from the original data set, and for each D_i train one classifier $h_{D_i}(x)$. Then average the hypothesis:

$$\hat{h}(x) = \frac{1}{M}\sum_{i=1}^{M} h_{D_i}(x)$$

It reduces the classifier's variance (with an increase in bias).

4. The Riddler uses linear regression (ordinary least squares) on a low dimensional data set. He is trying to reduce the classifier's bias with *boosting*. It is easy to change OLS to incorporate weights – but strangely he observes that his training error does not decrease even after many iterations. Write down the expression of the final boosted classifier after T iterations. What shape does the boosted decision boundary have? Can you explain his findings? [6]

Let $h_i(\vec{x}) = \vec{w_i}\cdot\vec{x}, i = 1, 2, ...$ be the set of weak learners the Riddler gets, and α_i be the corresponding weights. The boosting results will be

$$\hat{h}(x) = \sum_i \alpha_i h_i(\vec{x}) = \sum_i \alpha_i \vec{w_i}\cdot\vec{x} = \vec{w'}\cdot x,$$

where $\vec{w'} = \sum_i \alpha_i \vec{w_i}$. (A linear combination of hyperplanes is again a hyperplane.) The boosted regressor is still linear. The accuracy does not improve because OLS already returns the linear regressor that minimizes the squared loss. Boosting makes no difference here.

5. Linear Models

[20 Points]

1. Given a linear classifier $h_{w,b}(x) = sign(w^\top x + b)$. Show how a change of variable makes the explicit treatment of b unnecessary. Write down a new definition h_w. [2]

$w = \begin{bmatrix} w \\ b \end{bmatrix}$, $x' = \begin{bmatrix} x \\ 1 \end{bmatrix}$, $h_{w'}(x) = sign(w'^T x)$

2. Define the separating hyperplane $\mathcal{H}$ in terms of h_w. [2]

$$\mathcal{H} : \{x : h_w(x) = 0\}$$

3. How would you change your function for regression? [2]

$$h_w(x) = w^T x$$

4. Batgirl has n data points $(x_1, y_1), \dots, (x_n, y_n)$ with real valued labels $y_i \in \mathcal{R}$. She uses the following loss function to learn w:

$$w = \arg\min_w \sum_{i=1}^{n} (w^\top x_i - y_i)^2 + \lambda w^\top w.$$

   (a) Do you recognize the loss function (ignoring the $\lambda w^\top w$ for now)? For what data sets might this choice of loss function become problematic? [3]

   Squared Loss.
   When data sets contain a lot of noise.

   (b) Why would she add the term $\lambda w^\top w$? For what kind of data could this term for $\lambda > 0$ improve the test error? [3]

   L_2 regularization to avoid overfitting.
   Small and high dimensional data sets tend to overfit even with linear models.

   (c) Derive a closed-form solution for the vector w (You can use: $\frac{\partial trace(w^\top A)}{\partial w} = A$, $\frac{\partial trace(w^\top Bw)}{\partial w} = Bw + B^\top w$ and $w^\top w = w^\top I w$ where I is the identity matrix). [8]

   Let $X = [X_1, X_2, \cdots, X_n]^T, Y = [y_1, y_2, \cdots, y_n]^T$,

$$\begin{aligned} w &= \arg\min_w (Xw - Y)^2 + \lambda w^T w \\ &= \arg\min_w w^T X^T X w - w^T X^T Y - Y^T X w + Y^T Y + \lambda w^T w \\ &= \arg\min_w trace(w^T X^T X w - w^T X^T Y - Y^T X w + Y^T Y + \lambda w^T w) \end{aligned}$$

   Let $L = Tr(w^T X^T X w - w^T X^T Y - Y^T X w + Y^T Y + \lambda w^T w)$

$$\frac{\partial L}{\partial w} = 2X^T X w - 2X^T Y + 2\lambda I w = 0 \Rightarrow w = (X^T X + \lambda I)^{-1} X^T Y$$

Space for calculations or doodling.