CSE517a Midterm

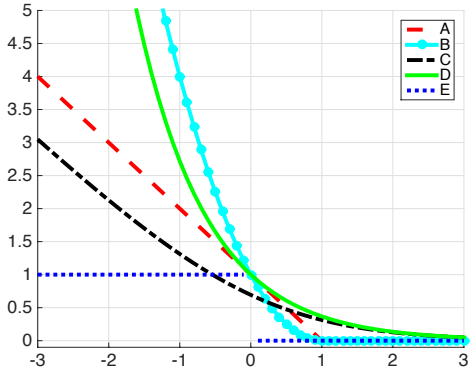
P

February 2015

NAME:	
Student ID:	
Email:	

GML	
kNN	
CART	
Bias Variance	
Boosting Bagging	
ERM	
TOTAL	

1 [15] ERM and SVM



1. (3) Write down the loss and regularizers of *SVM*, *LASSO* and *Ridge Regression*. Provide names for all loss functions and regularizers.

2. (2.5) Match the loss to the figure (just write A, ..., E next to each loss function's name.)

   - Zero-One Loss=
   - Hinge Loss=
   - Squared Hinge Loss=
   - Logistic Loss=
   - Exponential Loss=

3. (3.5) Which loss functions in Q2 could you minimize with Gradient Descent without modifications? Which could you minimize with Newton's Method? Justify your claims if you cannot use an approach for a given loss functions (no justification needed if a method *does* apply).

4. (1) You want to train an SVM to classify your data, but you also want to obtain a *sparse* weight vector. How would you modify the objective to achieve this goal?

5. You train a logistic regression classifier with gradient descent. Let $g(\vec{w})$ denote the gradient of the loss ℓ with respect to $\vec{w}$ (you don't need to compute it.)

   a) (2) State the update you are making from w_t to w_{t+1}.

   b) (4) Under what assumptions does this update reduce the loss? Prove your claim.

2 [??] General Machine Learning

Either circle **T** or **F**. Questions declared as **True** require no explanation (worth 1 point). Questions declared as **False** require a **one sentence** explanation (worth 2 points).

T/F Decreasing the depth of your decision tree (through pruning) will reduce test error.

T/F In k-fold cross validation you leave k inputs out, train your classifier on the remaining $n - k$ inputs and evaluate it on the leave-out inputs. You do this repeatedly and average to obtain a good estimate of your classifier's performance.

T/F The Adaboost algorithm stops when the training error is zero. This happens after a $O(log(n))$ iterations.

T/F Gradient Boosting is performing (stage-wise) gradient descent in function space.

T/F When you split your data into train and test you have to make sure you *always* do the splitting uniformly at random.

T/F If a classifier obtains 0% training error it cannot have 100% testing error.

T/F Increasing regularization tends to reduce the bias of your classifier.

T/F If run without depth limit, the ID3 algorithm returns the maximally compact decision tree that is consistent with a data set (if it exists).

T/F The best classifiers make no assumptions about your data at all.

T/F Random Forests learn many high variance CART trees and reduce this variance by averaging the results. That's basically Bagging applied to (slightly modified) CART trees.

T/F As your training data set size, n, approaches infinity, the $k-$nearest neighbor classifier is guaranteed to have an error no worse than twice the Bayes optimal error.

T/F Squared loss regression trees require a time complexity $O(n^2)$ per split.

T/F The Bayes optimal error is the best classification error you could get if there was no noise.

3 [15] Bias Variance

1. (3) Scrooge McDuck trains a classifier, trying to predict stock market prices. He trains decision trees of limited depth d (as he wants to reduce CPU time and electricity cost). However, soon he realizes that his training and his testing errors are both much too high for his system to be useful. Name **two** possible explanations for what could be the root of the problem.

2. (3) Scrooge now decides to no longer limit the tree depth. Instead he trains trees with unlimited depth. To his big disappointment he observes very similar behavior (train and test error are too high). What explanation can you give him?

3. (6) What does Bagging simulate and why would it reduce variance?

4. (3) You boost decision trees with very limited depth ($depth = 2$). How are bias and variance affected as the boosting iterations increase.

4 [17] kNN / Curse of Dimensionality

1. (3) Kim K. uses kNN classification with the Euclidean distance, *i.e.* $dist(\vec{x}, \vec{z}) = \sqrt{\sum_{i=1}^{d}(\vec{x}_i - \vec{z}_i)^2}$. She has $n = 100000$ data points in her training set, each with $d = 50$ dimensions. She is frustrated, because during test-time her classifier is too slow. What would you recommend her to do in order to speed up her classifier during test time?

2. She completely ignores your advice and instead analyses the code herself. She realizes that most of her computation time is spent computing the $\sqrt{}$ operator. Annoyed by this wasted CPU power, she decides to use the *squared distance* instead, $[dist(\vec{x}, \vec{z})]^2$ (and drops all square-root computations). Although faster to compute, this squared Euclidean distance is no longer a metric, as it does not satisfy the triangular inequality.

   a) (3) Will this missing property affect her kNN accuracy? Explain why/ why not.

   b) (3) If she is determined to use the squared distance, is your answer to question 1 still valid? Explain why/why not.

3. (3) Kanye W. wants to use the kNN classifier on images of dresses to classify them as either *white with gold stripes* (+1) or *blue with black stripes* (-1). The data set is rather high dimensional $d = 1000000$ (each feature is a pixel) and he only has about $n = 5000$ images. Should he be worried about the curse of dimensionality? Explain why/why not.

4. (2) Explain how the neighborhood size k affects the classifier's **bias** and **variance**.

5. (3) Provide one scenario in which you would want to use a kNN classifier instead of a linear SVM classifier and vice versa.

5 [9] Decision Trees

1. (3) What is the prediction value at a leaf of a regression tree (with squared-loss impurity)? Prove that this is optimal.

2. (2) Given the definition of the Gini Index of a set:

$$G(S) = \sum_{k=1}^{c} p_k(1 - p_k) \tag{1}$$

   where c is the total number of classes, and p_k is the probability that an item of set S is of class k. In the situation where there are 3 classes, when is G(S) maximized, when minimized? (no derivation necessary)?

3. (2) Explain in what sense decision trees are "myopic".

4. (2) What are two methods of preventing over-fitting in decision trees?

6 [15] Boosting Bagging

1. (4) Ludwig van Beethoven claims that when he trains a Random Forests classifier he does not need any validation or test data and can train the classifier on the entire data set, even to select model parameters? Is this true/false? Justify your answer.

2. (5) What is the loss function that Adaboost minimizes? Prove that it is an upper bound on the training error.

3. (6) Boosting chooses the next weak learner $h \in \mathcal{H}$ to maximize the inner-product with the gradient of ℓ w.r.t. the ensemble classifier $H(\vec{x}) = \sum_{t=1}^{T} \alpha_t h_t(\vec{x})$

$$h = \text{argmax} \sum_{i=1}^{n} \frac{\partial \ell(H)}{\partial H(x_i)} h(x_i)$$

In AdaBoost (with $y_i \in \{-1, 1\}$) this can be viewed as re-weighting the examples in each iteration. In Gradient Boosting (with $y \in \mathcal{R}$) this can be viewed as fitting the residual error with a squared loss. Derive **one of these two** updates. *(There is **no extra credit** for deriving both.)*