CS4780/5780 Final

Fall 2018

Instructions:

1. **Please turn off and stow away all electronic devices**
2. This is a **closed book and notes** exam.
3. You have **150 minutes**.
4. Please **don't** write on the back of the pages.
5. Write down individual steps to maximize for partial credit. If you believe a question is unclear, then please don't hesitate and ask us about it! Please always state any assumption you make. Good luck!

I promise to abide by Cornell's Code of Academic Integrity.

NAME:	
Net ID:	
Email:	

1 [??] General Machine Learning

Please identify if these statements are either True or False. Please justify your answer **if false**. Correct "True" questions yield 1 point. Correct "False" questions yield two points, one for the answer and one for the justification.

1. (BONUS: 3 pts) I filled out the course evaluation for CS4780/5780. (True/False, no explanation required).

2. (**T/F**) The fewer assumptions an algorithm makes, the better it is. In practice the best algorithm is Genetic Programming which makes no assumptions at all.

   False, all algorithms make assumptions

3. (**T/F**) With Random Forests there is no need to perform a training/validation split.
   True.

4. (**T/F**) MLE is great to learn the parameters of a binomial distribution, but it cannot be used to learn the parameters of a separating hyper-plane.
   False, the logistic loss in Logistic Regression is derived through MLE to learn the best separating hyperplane.

5. (**T/F**) The Naive Bayes classifier assumes that all features are independent.
False, It assumes all features are conditionally independent - given the label.

6. (**T/F**) Logistic Regression converges whenever a separating hyper-plane exists, otherwise it may run forever.
False. Logistic regression solves a convex optimization problem and always converges.

7. (**T/F**) The set of Support Vectors are all the the training data points an SVM cannot classify correctly.
False, they also include all training points with a margin of $<= 1$.

8. (**T/F**) A learned kernel SVM model (with RBF kernel) requires you to store some of the training data.
True (the support vectors)

9. (**T/F**) The decision boundary of a dual SVM classifier with linear kernel is identical to that of a primal SVM classifier.
True.

10. (**T/F**) l1 regularizer encourage sparse solutions.
True.

11. (**T/F**) In SVMs l2 regularization minimizes the squared bias term b^2.
False, the bias term is not regularized.

12. (**T/F**) Linear classifiers have as parameters the hyper-plane normal $\mathbf{w}$ and a bias term b. Reducing this bias term b will often increase the variance of the classifier.
False, the bias term is different from the bias/variance trade-off.

13. (**T/F**) The conditional distribution $P(y|\mathbf{x})$ of Gaussian Process Regression is itself a Gaussian distribution.
True.

14. (**T/F**) Kernelized linear regression (with RBF kernel) is a non-parametric algorithm.
True.

15. (**T/F**) A CART tree, if learned to full depth, are non-parametric algorithms.
True.

16. (**T/F**) In bagging, each classifier in the ensemble is trained on a data set that is independently and identically distributed.
False, the data is not independently sampled.

17. (**T/F**) One advantage of bagging is that all ensemble members (i.e. classifiers) can be trained in parallel.
True.

18. (**T/F**) AdaBoost with decision trees (depth 3) is non-parametric.
False, the set of parameters is not a function of the number of training instances, n.

19. (**T/F**) AdaBoost terminates the moment it reaches 0% training error.
False, as long as there is a weak learner with < 0.5 weighted training error, AdaBoost keeps boosting.

20. (**T/F**) One advantage of Random Forests is that you obtain meaningful probability estimates as your output predictions $P(y|\mathbf{x})$.
True.

21. (**T/F**) Deep convolutional neural networks are particularly well suited for image classification tasks.
True.

22. (**T/F**) The optimization of deep neural networks is a convex minimization problem.
False, it is non-convex because of the non-linear transition functions.

2 [19] Bias Variance / Model Selection

1. (3) Your Decision Tree classifier has a training error of 0% and a testing error of 87%. What can you say about the bias/variance trade-off (assuming the data is not noisy). Name two possible interventions to reduce the testing error? High Variance, Low Bias. You could prune the tree, or use bagging.

2. (3) For k-fold cross validation, describe the positive and negative effects as $k \to n$. When would you be most inclined to use $k = n$? The error decreases (as you have more training data) but as $k \to n$ the validation procedures also becomes very slower. You would use $k = n$ if you have very little training data (e.g. $n = 20$).

3. (6) The expected regression error decomposes into three terms. Write down the mathematical decomposition and label each term.

$$\underbrace{E_{\mathbf{x},y,D}\left[(h_D(\mathbf{x}) - y)^2\right]}_{\text{Expected Test Error}} = \underbrace{E_{\mathbf{x},D}\left[\left(h_D(\mathbf{x}) - \bar{h}(\mathbf{x})\right)^2\right]}_{\text{Variance}} + \underbrace{E_{\mathbf{x},y}\left[(\bar{y}(\mathbf{x}) - y)^2\right]}_{\text{Noise}} + \underbrace{E_{\mathbf{x}}\left[\left(\bar{h}(\mathbf{x}) - \bar{y}(\mathbf{x})\right)^2\right]}_{\text{Bias}^2}$$

4. (3) Explain why adding more training data does not always help reduce your testing error below a desired threshold $\epsilon > 0$. Describe such a scenario. The training error is a lower bound on the testing error. Adding more data increases the training error. If your *training* error is already too high ($> \epsilon$) adding more data will not help bring the testing error below ϵ as it is bounded by the training error.

5. (4) Consider the following algorithms and highlighted hyper-parameters. Decide whether *increasing* these parameters could help reduce overfitting. Answer with "Yes" or "No" ("Ja", and "Nein" is also acceptable.)

   a. The number of hidden units in the Neural Network.

   b. The maximum depth in Decision Trees.

   c. λ in Logistic Regression, trained with a $\lambda \sum_j w_j^2$ penalty in the objective.

   d. The number of iterations T in Boosting.

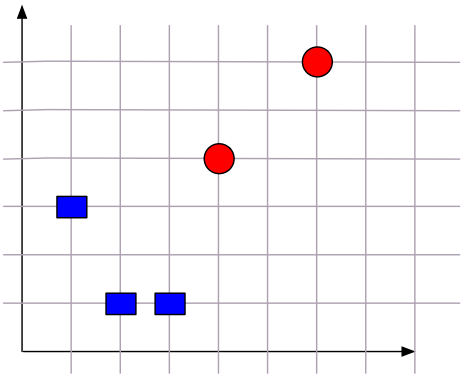
   a. Nein; b. Nein; c. Ja; d. Nein

3 [22] Kernel Methods

1. (2) Name one condition that is necessary and sufficient for a matrix $\mathbf{K}$ to be positive semi-definite. $\forall\ \mathbf{q}, \mathbf{q}\top\mathbf{K}\mathbf{q} \geq 0$ or $K = L\top L$ for some real matrix L, or K only has non-negative eigenvalues.

2. (3) Which of the following algorithms can be kernelized: a) Decision Trees, b) Linear Regression, c) Gaussian Processes. Justify your answer. b) and c) not a). b) and c) access data points only through inner-products, whereas a) splits on feature values and needs the feature realization of the data.

3. (4) Consider the following data set. Draw the decision boundary you would obtain with a hard margin linear SVM? Circle all the support vectors!

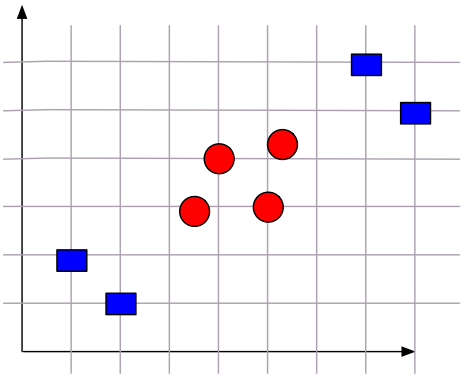


4. (3) Add two blue points (#1 and #2) such that #1 would and #2 *would not* affect the decision boundary if the SVM was re-trained.

5. (4) Let m be the number of support vectors of an SVM trained on n data points (with RBF kernel). For a fixed n imagine you increase the dimensionality d of the data until it becomes very large. How would you expect the ratio $\frac{m}{n}$ to change as $d \gg 0$? It approaches 1 because of the curse of dimensionality. All training points will be very far away from each other and close to the decision boundary.

6. (2) Describe a scenario in which you may want to use a kernel SVM with linear kernel instead of a standard linear (primal) SVM. If your dimensionality is very large, once the kernel is computed the computational complexity of kernel SVMs is independent of d.

7. (2) Consider the following data set. Draw a plausible decision boundary for a hard-margin SVM with polynomial kernel.



8. (2) You are given a non-linear regression data set. You are deciding between training a Gaussian Process or kernelized linear regression (both with RBF Kernel). Which one will have lower testing / training error? They are identical.

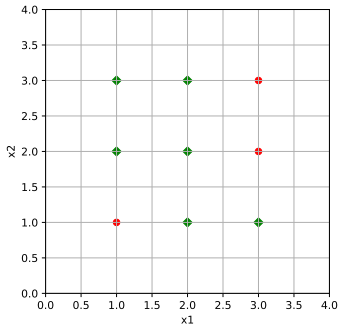
4 [14] Decision Tree

1. (4) Name two advantages of decision tree over nearest neighbor algorithms.
   We could select two of them:

   - once the tree is constructed, the training data does not need to be stored. Instead, we can simply store how many points of each label ended up in each leaf - typically these are pure so we just have to store the label of all points.
   - decision trees are very fast during test time, as test inputs simply need to traverse down the tree to a leaf - the prediction is the majority label of the leaf.
   - decision trees require no metric because the splits are based on feature thresholds and not distances.

2. (2) Name the CART stopping criteria (with unlimited depth). all labels are identical or all features are identical

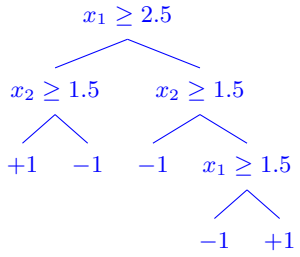
3. (8) Consider the classification dataset S with $|S| = 9$ visualized in the following figure and table:



i	$\mathbf{x}_i$	y_i
1	(1, 1)	+1
2	(1, 2)	-1
3	(1, 3)	-1
4	(2, 1)	-1
5	(2, 2)	-1
6	(2, 3)	-1
7	(3, 1)	-1
8	(3, 2)	+1
9	(3, 3)	+1

   (a) (2) Compute the *Gini impurity* for this dataset *before* any split. *Gini impurity*: $I_G(S) = \frac{1}{3} * \frac{2}{3} + \frac{2}{3} * \frac{1}{3} = \frac{4}{9}$.

(b) (6) Perform the CART algorithm with *Gini impurity* on S. Please draw a resulting tree (with splitting values and features) and also draw the corresponding hyper-planes in the previous figure.



5 [21 points] Ensemble Methods

1. (3) What loss function does AdaBoost minimize? (Write down the precise mathematical form.) The exponential loss $\frac{1}{n}\sum_{i=1}^{n} e^{-y_i H(x_i)}$ (the $\frac{1}{n}$ is optional).

2. (2) Imagine 10% of your binary training data (all points unique) are accidentally mislabeled. What is the training error that AdaBoost will converge to after sufficient rounds of boosting? 0%

3. (2) Describe a data scenario in which AdaBoost is not a good choice. Justify your answer. If you exhibit label noise. The exponential loss will ensure that the mislabeled data points will also be classified correctly and the algorithm will overfit (badly).

4. (6) Given a distribution P you can sample a training set D and obtain a classifier h. Imagine you train m such classifiers $h_1, \ldots, h_m$ on m data sets $D_1, \ldots, D_m$, each drawn i.i.d. from the data distribution P. As you increase m from $m = 1$ to $m \gg 0$,

   - show how you can use these models to obtain a low variance classifier $\hat{h}$.
   - what happens to the variance of $\hat{h}$ in the limit, $m \gg 0$?
   - how does the bias of $\hat{h}$ compare to the bias of h?

   You average them: $\hat{h} = \frac{1}{m}\sum_{i=1}^{m} h_i$. By the weak law of large numbers the average $\hat{h}$ will approach the expected classifier $\bar{h}$ as $m \gg 0$ and $E_{\mathbf{x},D}\left[\left(h_D(\mathbf{x}) - \bar{h}(\mathbf{x})\right)^2\right] \to 0$. The bias is unaffected, i.e. the bias of $\hat{h}$ is identical to the bias of h, because the $E[\hat{h}] = E[h]$.

5. (4) After two iterations of AdaBoost, with step sizes α_1, α_2 respectively and weak learners h_1, h_2, what are all possible weights that could potentially be assigned to a training data point (ignore normalization). $e^{-\alpha_1-\alpha_2}, e^{-\alpha_1+\alpha_2}, e^{+\alpha_1-\alpha_2}, e^{\alpha_1+\alpha_2}$

6. (4) Robin is trying to use AdaBoost on full CART trees without depth limit (all training points are distinct). Although the code seems correct, it crashes in the very first round. What do you think is the problem? The CART tree has zero classification error, yielding an infinite step-size $\alpha = \frac{1}{2}\ln(\frac{1-\epsilon}{\epsilon})$ and a division by zero.

6 [12] Deep Learning

1. (2) Name two reasons why Newton's Method typically is not used to train deep neural networks. 1. too many parameters to store the Hessian; 2. it converges quickly to the closest local minima / saddle point and not to a wide minimum

2. (2) Let the loss function be $\ell(\mathbf{w}) = \frac{1}{2n}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w} - y_i)^2$. Write down the update for *Stochastic* Gradient Descent and Gradient Descent. $G_{GD} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i^\top \mathbf{w} - y_i)\mathbf{x}_i$ whereas the SGD update is $G_{SGD} = \frac{1}{m}\sum_{i=1}^{m}(\mathbf{x}_{s_i}^\top \mathbf{w} - y_{s_i})\mathbf{x}_{s_i}$ for randomly picked $s_i \in [n]$.

3. (2) Suppose you have a convolutional filter of size $k \times k$. When you apply this filter to a $n \times n$ input image, what is the dimension of the output feature map with no padding?
$(n-k+1) \times (n-k+1)$

4. (2) Suppose you have a 3×3 matrix I from one patch of an image. Each matrix value corresponds to a pixel.

$$I = \begin{bmatrix} 3 & 1 & 1 \\ 3 & 0 & 2 \\ 4 & 4 & 0 \end{bmatrix}$$

and filter kernel

$$k = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

What is the output matrix after convolving the input I with k (no flipping of the kernel in case you learned that in your computer vision/signal processing class)? We don't consider the padding and stride here. The output should be a

2×2 matrix.

$$\begin{bmatrix} 6 & 3 \\ 11 & 4 \end{bmatrix}$$

5. (4) Consider you have the following neural network:

   - Input layer: 80 units
   - First hidden layer: 20 hidden units
   - Second hidden layer: 60 hidden units
   - Third hidden layer: 20 hidden units
   - Output layer: 80 units
   - Sigmoidal activation for each hidden layer and the output
   - Loss function: logistic loss

   Each layer has a bias. How many parameters does this neural network have? You can leave your answer as an expression.

   $$\#\text{ params} = 80 \cdot 20 + 20 \cdot 60 + 60 \cdot 20 + 20 \cdot 80 + 20 + 60 + 20 + 80 = 4780$$

   . Please remember 4780, our course!

This page is left blank for jokes. (Nothing too inappropriate.)

This page is left blank for scratch space.