Midterm Solutions: CS 6375
Fall 2019

The exam is closed book (1 page cheat sheet allowed). Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, use an additional sheet (available from the instructor), write your name on the sheet and staple it to your exam.

- **NAME** ____________________
- **UTD-ID if known** ____________________

Question	Points	Score
Decision Trees	10	
Neural Networks	10	
Support Vector Machines	10	
Short Questions	10	
AdaBoost	10	
Total:	50	

Question 1: Decision Trees (10 points)

(a) (5 points) Is the following statement True/False. Justify your answer. No credit if the justification/explanation is incorrect.

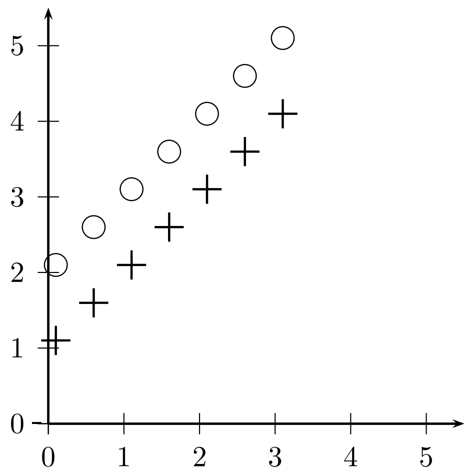
The decision tree algorithm we discussed in class (the one using the information gain heuristic without pruning) yields a globally optimal decision tree. (By optimal, we mean that it yields a decision tree having the smallest number of nodes and the highest accuracy on the training set.)

Solution: False. The decision tree algorithm we discussed in class is a greedy algorithm. It does not perform "backtracking" which will be needed to yield a globally optimal solution.

(b) (5 points) Is the following statement True/False. Justify your answer. No credit if the justification/explanation is incorrect.

You are given a d-dimensional linearly separable training data having n examples. The size of the decision tree for this data is guaranteed to be polynomial in d.

Solution: False. The size of the decision tree can be polynomial in n even when d is constant. For example, consider the two dimensional dataset given below.



$O(n)$ nodes will be required here. Since all splits are either horizontal or vertical, each of them will classify at most one point correctly and have roughly equal number of positive and negative examples in each partition of the data after the split.

Question 2: Neural Networks (10 points)

(a) (6 points) Draw a neural network having minimum number of nodes that represents the following function. Please provide a precise structure as well as a setting of weights. You can only use **simple threshold units** (namely, $o = +1$ if $\sum_i w_i x_i > 0$ and $o = -1$ otherwise) as hidden units and output units. X_1, X_2 and X_3 are attributes and Y is the class variable.

X_1	X_2	X_3	Y
0	0	0	$+1$
0	0	1	-1
0	1	0	$+1$
0	1	1	$+1$
1	0	0	$+1$
1	0	1	-1
1	1	0	$+1$
1	1	1	$+1$

Solution: This can be represented using a Perceptron (having a simple threshold unit as its output unit). The concept is $X_2 \vee \neg X_3$.
Weights used: $w_0 = 0.5$, $w_1 = 0$, $w_2 = 1.0$ and $w_3 = -1$.

Notes: Modeling the truth table as CNF, we have $(\neg X_1 \vee X_2 \vee \neg X_3) \wedge (X_1 \vee X_2 \vee \neg X_3)$. Resolving the first clause with the second clause we get $X_2 \vee \neg X_3$. Therefore, the CNF is equivalent to $(\neg X_1 \vee X_2 \vee \neg X_3) \wedge (X_1 \vee X_2 \vee \neg X_3) \wedge (X_2 \vee \neg X_3)$. Via subsumption elimination (namely eliminating clauses which are subsumed by another clause), we can remove the first and the second clause because they are subsumed by the third. Therefore, the CNF is $X_2 \vee \neg X_3$.

(b) (2 points) True/False. Explain your answer in 1-2 sentences. Given a neural network having one hidden layer and at least n hidden nodes where n is the number of features, the back-propagation algorithm is susceptible to initialization. Namely, the parameters returned by the algorithm will be different for different initialization strategies.

Solution: True. Backpropagation on such networks reaches a local minima and typically there are a large number of them. Therefore, different initialization strategies will likely yield different parameters.

(c) (2 points) Describe one approach to prevent over-fitting in neural networks.

Solution: See class slides. (1) Early Stopping using a validation set; (2) L_2 penalty; etc.

Question 3: Support Vector Machines (10 points)

Consider the dataset given below (x_1, x_2 are the attributes and y is the class variable):

x_1	x_2	y
0	0	-1
$-1/3$	1	$+1$
$1/3$	1	$+1$
-2	-2	$+1$

(a) (3 points) Write the expression for the primal problem for this dataset under the assumption that you are using the linear SVM.

Solution: Primal problem:

$$\text{minimize } L(\mathbf{w}, \lambda, b) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{4} \lambda_i (y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1)$$

subject to $\lambda_i \geq 0$, where i indexes over the examples.

For the given dataset, we have:

$$L(\mathbf{w}, \lambda, b) = \frac{1}{2}(w_1^2 + w_2^2) + (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) +$$
$$\lambda_1(b) - \lambda_2(-\frac{1}{3}w_1 + w_2 + b) - \lambda_3(\frac{1}{3}w_1 + w_2 + b) - \lambda_4(-2w_1 - 2w_2 + b)$$

(b) (2 points) Is the data linearly separable? Circle one: YES NO.

Solution: The dataset is linearly separable. Plot the data!

(c) (5 points) If your answer to the previous question is NO, please suggest a suitable Kernel which will ensure separability (namely, the function used separates the positive examples from the negative examples). If your answer is YES, please use the linear Kernel for this part. Write the expression for the SVM dual problem w.r.t. the Kernel and the data given above.

Solution: We will use a linear kernel since the data is linearly separable. The dual problem is:

$$\text{maximize} \sum_{i=1}^{4} \lambda_i - \frac{1}{2} \sum_{i=1}^{4} \sum_{j=1}^{4} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

such that $\lambda_i \geq 0$ and $\sum_{i=1}^{4} \lambda_i y_i = 0$.

For the given dataset we have:

The term $y_i y_j \mathbf{x}_i^T \mathbf{x}_j$ for different values of i and j in the double sum is given by:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 10/9 & 8/9 & -4/3 \\ 0 & 8/9 & 10/9 & -8/3 \\ 0 & -4/3 & -8/3 & 8 \end{bmatrix}$$

The constraints are: $\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0$ and $\lambda_2 + \lambda_3 + \lambda_4 - \lambda_1 = 0$.

Question 4: Short Questions (10 points)

(a) (3 points) Assume that you are given a Naive Bayes model defined over n binary features and one class variable having two values. Assume that at test time, k out of the n features are missing. Then, is the following statement True or False. *The Naive Bayes model will be impractical because determining the posterior marginal probability distribution over the class variable given the observed features will require time that scales exponentially in k in the worst case*. Explain your answer.

Solution: False. Marginal probability of the class variable is given by

$$P(c|x) \propto \sum_{m} P(c,x,m) = \sum_{m_1,\ldots,m_k} P(c) \prod_{i=1}^{n-k} P(x_i|c) \prod_{j=1}^{k} P(m_j|c)$$

Simplifying, we get:

$$P(c|x) \propto P(c) \prod_{i=1}^{n-k} P(x_i|c) \left(\sum_{m_1,\ldots,m_k} \prod_{j=1}^{k} P(m_j|c) \right)$$

We can commute the sum and product in the bracket because each sum depends only on one variable. Therefore, we get:

$$P(c|x) \propto P(c) \prod_{i=1}^{n-k} P(x_i|c) \prod_{j=1}^{k} \sum_{m_j} P(m_j|c)$$

When we sum out m_j, we get a 1 because $P(m_j|c)$ is a probability distribution. Therefore, the above equation simplifies to:

$$P(c|x) \propto P(c) \prod_{i=1}^{n-k} P(x_i|c)$$

Thus, the complexity of inference (computing the posterior marginal distribution over the class) is actually linear in the number of observed features.

(b) (3 points) *The bias of a k nearest neighbor classifier increases with k*. True/False. Explain your answer.

Solution: True. Increasing k will decrease variance and increase bias because the functions become smoother and less susceptible to change if we add or delete data points. In fact when k equals the number of data points, kNN represents a simple function because our prediction for any test point will be the average over all training data points if we are solving a regression problem or the majority class in the training data if we are solving a classification problem.

(c) (4 points) Let us assume that the data (y) was generated from the following distribution:

$$\Pr(y) = \frac{\theta^y e^{3.5\theta}}{y!}$$

Let us assume that you are given n data points $y_1, \ldots, y_n$ drawn independently from $\Pr(y)$.

Write down the expression for the log-likelihood of data.
Derive an expression for θ such that the log-likelihood of data is maximize (namely find the maximum likelihood estimate of θ).

Solution: Consider the expression for the log-likelihood of the data:

$$LL = \sum_{i=1}^{n} \log(\Pr(y_i)) = \sum_{i=1}^{n} \log\left(\frac{\theta^{y_i} e^{3.5\theta}}{y_i!}\right)$$

Simplifying the term inside the bracket, we get

$$LL = \sum_{i=1}^{n} y_i \log\theta + 3.5\theta - \log(y_i!)$$

The maximum likelihood estimate of θ is given by:

$$\theta_{MLE} = \arg\max_{\theta} \left(\sum_{i=1}^{n} y_i \log\theta + 3.5\theta - \log(y_i!)\right)$$

We can solve this by computing the derivative of the log-likelihood w.r.t. θ and setting it to zero.
The derivative is given by

$$\sum_{i=1}^{n} \frac{y_i}{\theta} + 3.5$$

Setting it to zero and solving for θ, we get

$$\theta_{MLE} = -\frac{\sum_{i=1}^{n} y_i}{3.5n}$$

Question 5: AdaBoost (10 points)

Consider the following dataset. $(X_1, X_2) \in \mathbb{R}^2$ and Y is the class variable.

X_1	X_2	Y
0	8	$-$
1	4	$-$
3	7	$+$
-2	1	$-$
-1	13	$-$
9	11	$-$
12	7	$+$
-7	-1	$-$
-3	12	$+$
5	9	$+$

We will use two rounds of AdaBoost to learn a hypothesis for this data set. In each round m, AdaBoost chooses a weak learner that minimizes the error ϵ_m. As weak learners, use hypotheses of the form (a) $h_1 \equiv [X_1 > \theta_1]$ or (b) $h_2 \equiv [X_2 > \theta_2]$, for some integers θ_1, θ_2 (either one of the two forms, not a disjunction of the two). There should be no need to try many values of θ_1, θ_2; appropriate values should be clear from the data.

(a) (3 points) Which weak learner will AdaBoost choose in the first iteration ($m = 1$)? Be sure to provide a precise value for θ_1 or θ_2 for this learner.

Solution: For h_1, the error rates for all possible values of θ are as follows (format is θ: err):
$-7.5 : 0.6; -3.5 : 0.5; -2.5 : 0.6; -1.5 : 0.5; -0.5 : 0.4; 0.5 : 0.3; 1.5 : 0.2; 3.5 : 0.3; 5.5 : 0.4; 9.5 : 0.3; 12.5 : 0.4$
For h_2, the error rates for all possible values of θ are as follows (format is θ: err):
$-1.5 : 0.6; 0.5 : 0.5; 1.5 : 0.4; 4.5 : 0.3; 7.5 : 0.5; 8.5 : 0.4; 9.5 : 0.5; 11.5 : 0.4; 12.5 : 0.5; 13.5 : 0.4$

Since $h_1 \equiv [X_1 > 1.5]$ has the smallest error rate (0.2), Adaboost will choose $h_1 \equiv [X_1 > 1.5]$.

(I have copied the data from the previous page to this page for your convenience.)

X_1	X_2	Y
0	8	$-$
1	4	$-$
3	7	$+$
-2	1	$-$
-1	13	$-$
9	11	$-$
12	7	$+$
-7	-1	$-$
-3	12	$+$
5	9	$+$

(b) (7 points) Which weak learner will AdaBoost choose in the second iteration ($m = 2$)? Again, be sure to provide a precise value for θ_1 or θ_2 for this learner.

Solution: The new weights for the examples are given below:

$$0.1, 0.1, 0.1, 0.1, 0.1, 4.0, 0.1, 0.1, 4.0, 0.1$$

$h_1 \equiv [X_1 > 1.5]$ makes 2 errors. It misclassifies the 6^{th} and the 9^{th} examples. The weights of these examples are updated to $(1 - \epsilon_m)/\epsilon_m) = (1 - 0.2)/0.2 = 4.0$. For h_1, the error rates for all possible values of θ are as follows (format is θ: err):
$-7.5 : 0.5625; -3.5 : 0.5; -2.5 : 0.75; -1.5 : 0.6875; -0.5 : 0.625; 0.5 : 0.5625; 1.5 : 0.5; 3.5 : 0.5625; 5.5 : 0.625; 9.5 : 0.375; 12.5 : 0.4375$
For h_2, the error rates for all possible values of θ are as follows (format is θ: err):
$-1.5 : 0.5625; 0.5 : 0.5; 1.5 : 0.4375; 4.5 : 0.375; 7.5 : 0.5; 8.5 : 0.4375; 9.5 : 0.5; 11.5 : 0.25; 12.5 : 0.5; 13.5 : 0.4375$
(Note that I have not normalized the error rates. Normalized error rates can be obtained by dividing the error rate by $\sum_{i=1}^{10} w_i$.)
Since $h_2 \equiv [X_2 > 11.5]$ has the smallest error rate (0.25), Adaboost will choose $h_2 \equiv [X_2 > 11.5]$.