University of TEHRAN

Machine Learning- Spring 2020

Final Examination

Department of Bioinformatics, IBB, University of Tehran

July 21, 2020

Instructors: Hesam Montazeri and Kaveh Kavousi

240 minutes

Your name: ---------- (please *initial all pages!)*

Academic honor code

I attest that I have not given or received aid in this examination and completely uphold the principles and rules of the academic honor code.

Your name: **Signature**

You can use your handwritten class notes during the exam. Good luck!

Disclaimer: *Few questions are adapted from online resources.*

Please leave the below table empty.

	PROBLEM	MAXIMUM POINTS	OBTAINED POINTS
1	Short questions	15	
2	Ridge regression	12	
3	DNA sequence classification	10	
4	VC dimension	8	
5	Random forest or LASSO	10	
6	Boosting	20	
7	Euclidean distance & triangular inequality	5	
8	Hard and fuzzy clustering	12	
9	Neural networks for two-spiral task	8	
TOTAL		100	

1. Short questions (15 points)

i. Explain the relationship between MAP, MLE, and prior knowledge in general or in the context of an example. (3 pt)

ii. Why ReLU activation function is popular in neural networks? (3 pt)

iii. Cross validation is a standard way for choosing hyperparameter λ in lasso. Assume you can only afford to cross validate for 100 different λ values. How would be your strategy for choosing these values in order to find the optimal λ? (trying a grid of values is not a choice) (3 pt)

iv. Which of the following is the main reason for having weak learners in boosting algorithm? To reduce bias or to reduce variance. Explain! (3 pt)

v. How would you determine feature importance using neural networks? (3 pt)

2. Ridge regression (12 points)

The ridge regression optimization task on centered data is defined as

$$\min_{\beta} (y - X\beta)^T (Y - X\beta) + \lambda \beta^T \beta$$

where $\beta = (\beta_1, \dots, \beta_p)^T$. Your task is to

a) derive a coordinate descent algorithm for the parameter estimation of ridge regression (6 pt)

b) derive the updating rule of stochastic gradient descent for ridge regression using the Huber loss criterion instead of squared error loss. The Huber loss is defined as (6 pt)

$$L(y, f(x)) = \begin{cases} [y - f(x)]^2 & \text{for } |y - f(x)| \leq \delta, \\ 2\delta|y - f(x)| - \delta^2 & \text{otherwise.} \end{cases}$$

3. DNA sequence classification (10 points)

Consider the task of classifying a variable-length input sequences to either α-chain or β-chain of MHC II molecules. Some examples of the training observations are given as follows. (Note the data in the following example are synthetic and are given for illustration of the classification problem)

Input sequence	length	class
gcagttcagcctacccgctttaaggtgct……ccgtcac	248	α-chain
tcagttcagcctagcagctttaaggtgct……ccattat	263	β-chain
ggacgctcaagcatacacgctttaaggtgct……gcgttaa	278	α-chain

Assume the overall information in the sequences matters not their exact positions. Your task is to develop two classification models for this problem. Suppose you have a function that can generate all possible k-mers for a given sequence.

a) Design an appropriate kernel function for the support vector machine. (5 pt)

b) How can you use Naïve Bayes model for this task? Explain in detail. (5 pt)

4. VC dimension (8 points)

Compute the VC dimensions of the following models in 2D space (you need to provide a lower bound for the VC dimension and a counterexample or justification, if necessary). No formal proof is required.

I. Origin-centered circle. You can choose whether it is the inside or the outside of the circle which gets the class +1. (4 pt)

II. A circle not necessarily centered at origin. (4 pt)

5. Random forest or LASSO (10 points)

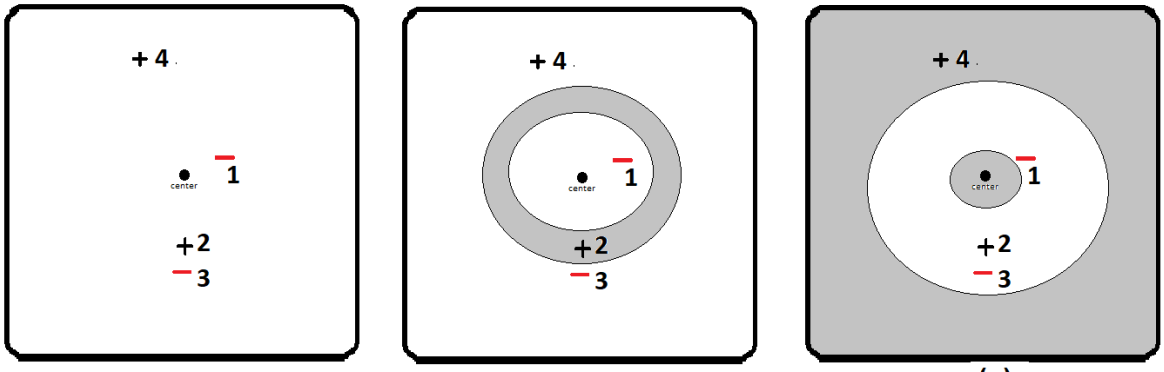
Given a large number of observations, the goal is to solve a binary classification task with 10 relevant and 10000 completely irrelevant input features. Assume the input data is balanced (50% chance for each class). You can use either use 1) LASSO (λ is chosen by cross validation) or 2) random forest with k=10 where k is the number of features selected at each split. In addition, each tree can have up to 100 splitting points.

a) Which model will you use for this setting? Explain why? (4 pt)

b) Provide reasonable lower and upper bounds for accuracy of the above random forest model (**hint:** a random classifier has 50% accuracy on a balanced dataset and a perfect classifier has 100% accuracy) (6 pt)

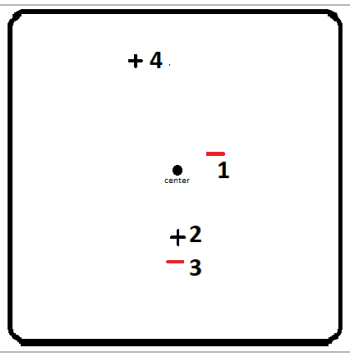
6. Boosting (20 points)

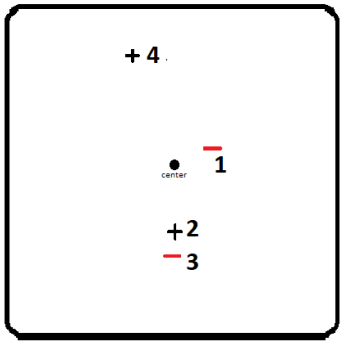
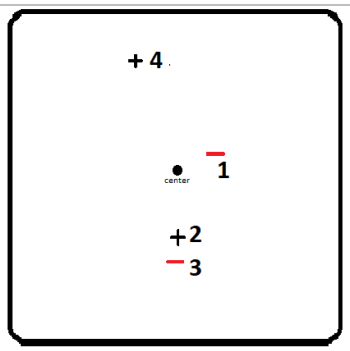
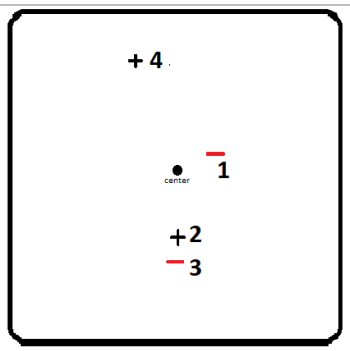
In this question, you need to investigate how Adaboost works for a binary classification problem with two concentric circles as a weak learner. This learner classifies the area between two circles as positive or negative class and the complementary area to the other class. Your task is to apply AdaBoost algorithm (see below) for training data shown in the part *a* of below figure; part *b* and *c* show two possible classifiers where highlighted areas indicate regions classified as positive.

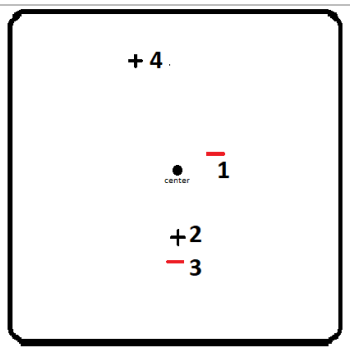


Algorithm 10.1 *AdaBoost.M1.*

1. Initialize the observation weights $w_i = 1/N,\ i = 1, 2, \ldots, N$.
2. For $m = 1$ to M:
   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i.
   (b) Compute
   $$\text{err}_m = \frac{\sum_{i=1}^{N} w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^{N} w_i}.$$
   (c) Compute $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$.
   (d) Set $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))],\ i = 1, 2, \ldots, N$.
3. Output $G(x) = \text{sign}\left[\sum_{m=1}^{M} \alpha_m G_m(x)\right]$.

a) Draw the decision boundary learned by G_1. Compute α_1 and weights. Circle misclassified points by G_1. (4 pt)	

b) Draw the decision boundary learned by G_2. Compute α_2 and weights. Circle misclassified points by G_2. (4 pt)	
c) Draw the decision boundary learned by G_3. Compute α_3 and weights. Circle misclassified points by G_3. (4 pt)	
d) Indicate how $G = sgn(\alpha_1 G_1 + \alpha_2 G_2 + \alpha_3 G_3)$ classify each of the four points? (4 pt)	

e) What is the minimum training error achievable by boosting for this example? Prove your point! (4 pt)	

7. Euclidean distance & triangular inequality (5 points)

Prove that the Euclidean distance satisfies the triangular inequality. *Hint*: Use the Minkowski inequality, which states that for a positive integer *p* and two vectors $x = [x_1, \dots, x_l]^T$ and $y = [y_1, \dots, y_l]^T$ it holds that:

$$\left(\sum_{i=1}^{l} |x_i + y_i|^p\right)^{1/p} \leq \left(\sum_{i=1}^{l} |x_i|^p\right)^{1/p} + \left(\sum_{i=1}^{l} |y_i|^p\right)^{1/p}$$

8. Hard and fuzzy clustering (12 points)

Suppose $X = \{x_1, x_2, x_3, x_4\}$ is given where $x_1 = [0,0]^T$, $x_2 = [2,0]^T$, $x_3 = [0,3]^T$, $x_4 = [2,3]^T$. Let $\theta_1 = [1,0]^T$ and $\theta_2 = [1,3]^T$ be the cluster representatives. In addition, the Euclidean distance between a vector and a representative is used as the distance measure. The hard two-cluster clustering that minimizes

$$J_q(\boldsymbol{\theta}, U) = \sum_{i=1}^{N}\sum_{j=1}^{m} u_{ij}^q d(\boldsymbol{x}_i, \boldsymbol{\theta}_j)$$

for the above choice of θ_1, θ_2, can be represented by

$$U_{hard} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

where *N* and *m* are the number of data points and clusters, respectively.

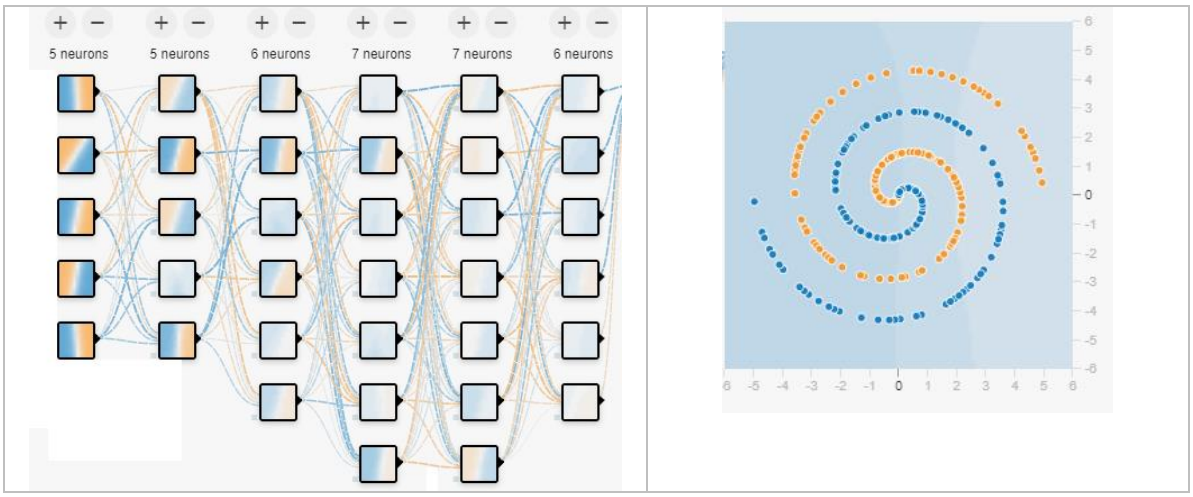
a) Obtain the value of $J_q^{hard}(\theta, U_{hard})$ (note in this case J does not depend on *q*) (4 pt)

b) Prove that for *m= 2, q =1* and for fixed θ, always the hard clustering is favored against the fuzzy ones (**Hint:** show that always $J_1^{fuzzy}(\theta, U) > J_q^{hard}(\theta, U)$) (4 pt)

c) Prove that for *m= 2, q =2* and for fixed θ, there are cases where the fuzzy clusterings are favored against the hard ones (**Hint:** you need to show the following formula is not always true $J_1^{fuzzy}(\theta, U) > J_q^{hard}(\theta, U)$) (4 pt)

9. Neural networks for two-spiral task (8 points)

The following neural network is used for a noisy two-spiral classification task (only hidden layers are displayed):



The activation function is tangent hyperbolic and L2 regularization rate is 0.003. The training and test errors are 0.27 and 0.39, respectively. Assume the Bayes error is 0.03. You can only apply **two of the following modifications** to find a better neural network. What would be your choices? Explain why.

i. Decrease/increase dropout rate
ii. Decrease/increase hidden layers and number of neurons
iii. Decrease/increase regularization rate
iv. Modify feature space by removing/adding some non-linear terms
v. Use ReLU activation function
vi. Use sigmoid activation function