CSE 847: Machine Learning Assignment #2

Instructor: Prof. Jun Wu
Out: Sep. 26, 2024; Due: Oct. 30, 2024
Submit electronically for Assignment #2, a file named `yourFirstName-yourLastName.pdf` *containing your solution to this assignment (a .doc or .docx file is also acceptable, but .pdf is preferred).*

1 Support Vector Machines [20 points]

SVM is a model that fits *the maximum-margin hyperplane* in a certain feature space. In real-world problems, it is difficult to know if a data set is separable or not, even with the kernel trick. Therefore, soft-margin SVM is introduced to handle those samples that could hardly be correctly classified.
1. [10 points] Please describe the difference between soft-margin SVM and hard-margin SVM, and explain why.

2. [10 points] After training a soft-margin SVM, we have three categories (not classes) of samples based on the value of the slack variable. Let's use ξ_i to denote the value of the slack variable for a certain sample $\mathbf{x}_i$, please write down how we categorize (not classify) a sample based on the value of ξ_i. Would the decision boundary change if such a sample is removed from the training set? Please justify your answer.

2 AdaBoost [40 points]

1. [20 points] Suppose you are given a **balanced** binary training data set, i.e., the number of data points labeled +1 equals the number of data points labeled -1.

(a). In such a balanced data set, what are the basic requirements of the Adaboost algorithm regarding the adopted weak classifiers?

(b). What will happen if we use weak binary classifiers whose classification accuracy is **less than** 50%, say 30%? Will the performance of the combined classifier improve with more iterations?

2. [20 points] Suppose you are given another simple training data set (different from the one used in the previous question) shown below consisting of 10 samples ($\mathbf{x}$, y) in Table 1. Here y is the desired class label for each corresponding $\mathbf{x}$.

$\mathbf{x}$	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

Table 1: The training set

For the weak classifier C, we will use a single threshold θ so that

$$C(x) = \begin{cases} +1 & x < \theta; \\ -1 & x \geq \theta. \end{cases}$$

(a). In the initialization phase, we let the initial threshold $\theta = 2.5$ which minimizes the classification error at Iteration 1. Show the weights over examples from iteration $2 \sim 6$, respectively.

(b). How do the weights of the training samples change after 4 iterations? Does including the fifth and sixth weak classifiers improve the performance of the combined classifier?

3 K-Nearest Neighbor Classifier [40 points]

1. [10 points] **A Lazy Classifier.** Considering the online learning situation where besides the observed training samples, we have new samples becoming available as time goes on, some classifiers have to be retrained from scratch.

(a). When a new training sample becomes available, among SVM, Naive Bayes and KNN, which classifier(s) have to be retrained from scratch? Please justify your answer.

(b). When a new test sample becomes available, among SVM, Naive Bayes and KNN, which classifier needs the most computation to infer the class label for this sample? What is the time complexity for this classifier (using $O(\cdot)$ notation) assuming we have n training samples? Please justify your answer.

2. [30 points] **Implementation of KNN Classifier.** Evaluate the KNN Classifier on the MNIST data set where each sample is a picture of a handwritten digit. Each sample includes 28x28 grey-scale pixel values as features and a categorical class label out of 0-9. You have to code a KNN classifier with *Euclidean distance* from scratch, **do not use existing class or functions (e.g. the function from sklearn.neighbors.KNeighborsClassifier)**

The processed MNIST dataset is provided in 'MNIST data.zip'. It contains two files 'train data.mat' and 'test data.mat' which represent the training set and the test set, respectively. For example, in 'train data.mat', $\mathbf{X} \in \mathbb{R}^{60000 \times 784}$ corresponds to the input feature matrix with each row being a data point's feature vector $\mathbf{x}$. $\mathbf{Y} \in \mathbb{R}^{60000 \times 1}$ corresponds to the column vector of labels. In your implementation, please use the **first** 6,000 samples from the orignal training set for training KNN, and the **first** 1,000 from the original test set for testing KNN.

(a). Briefly describe how you implement the KNN classifier by giving the pseudocode. The pseudocode must include equations regarding how the distances are computed and how classification is done for each sample in the test set. Remember, this should not be a printout of your code, but a high-level outline description. Include the pseudocode in your pdf file (or .doc/.docx file). Submit the actual code as a single zip file named yourFirstNameyourLastName.zip IN ADDITION TO the pdf file (or .doc/.docx file).

(b). Plot curves for training and test errors: the error rate (error rate = 1 - accuracy, y-axis) vs. the value of K (x-axis). Plot 11 points for the curve, using $K = 1, 9, 19, 29, 39, 49, 59, 69, 79, 89, 99$, using the same training set and test set each time. Average your results over 5 runs using each K (e.g., 99). Plot the error curves for training error and test error on the same figure.